LEARNING PERTURBATION-SPECIFIC REPRESENTATIONS THROUGH CONDITIONAL ENERGY PRIOR FOR CELLULAR RESPONSE PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

AI-based cellular perturbation prediction is essential to accelerate drug discovery and therapeutic identification. However, perturbation-specific signals are generally obscured by cellular complexity, making it difficult for existing methods to correctly capture perturbation effects. To address this problem, we propose PerturbEnergy, which introduces a conditional energy-based prior over perturbation representations. It assigns lower energy (higher probability) to representations compatible with the given perturbation and higher energy to others, thereby explicitly tilting the latent space toward perturbation-preferred regions. Besides, PerturbEnergy can be integrated into autoencoder architectures as we find its Evidence Lower Bound (ELBO) is equivalent to the ELBO of the complete autoencoder. We further prove that optimizing PerturbEnergy increases the lower bound on the conditional mutual information between representation and perturbation. Extensive experiments on four benchmark datasets demonstrate the effectiveness of our PerturbEnergy: it improves perturbation recognition especially in latent space with improvement of up to 83.2%; It also achieves state-of-the-art performance for predicting top-50/100/200 differentially expressed gene(DEG) response and remaining competitive on highly variable gene(HVG) prediction. The code is provided anonymously: https://anonymous.4open.science/r/AnonymousPerturbEnergy-F5D2

1 INTRODUCTION

Cellular perturbation prediction aims to forecast cell responses to drug and genetic interventions, supporting therapeutic target identification, drug discovery, and personalized medicine. While modern high-throughput CRISPR technologies Jinek et al. (2012); Gilbert et al. (2014); Dixit et al. (2016); Replogle et al. (2020) have dramatically increased the scale of single-cell transcriptomic profiling, experimentally screening the vast space of perturbations across diverse cellular contexts remains costly and labor-intensive. To address this challenge, AI-based virtual cells (AIVCs) Bunne et al. (2024); Resasco et al. (2012) have emerged as an efficient approach for predicting cellular responses using deep learning models.

Existing methods generally follow a representation learning paradigm Jiang et al. (2026); Russkikh et al. (2020) where the perturbation effect is captured within the latent representation space. Some approaches use foundation models Cui et al. (2024); Hao et al. (2024); Wang et al. (2024) or specialized encoders Lotfollahi et al. (2019); Russkikh et al. (2020); Jiang et al. (2026); Roohani et al. (2024) to encode cells into latent representations and subsequently predict perturbation responses through generative decoding. Other studies develop more powerful architectures to directly model perturbation-induced transitions between control and perturbed cell representations. For example, STATE Adduri et al. (2026) employs a transition Transformer to model perturbation-induced shifts between cell representations; CellFlow Klein et al. (2025) uses optimal transport to learn mappings between control and perturbed cell distributions; Squidiff He et al. (2026) and PerturbDiff Yuan et al. (2026) characterize such representation transitions through diffusion processes.

Although these methods have demonstrated effectiveness across multiple datasets, they overlook an important issue that may substantially limit their performance. Accurate perturbation prediction

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

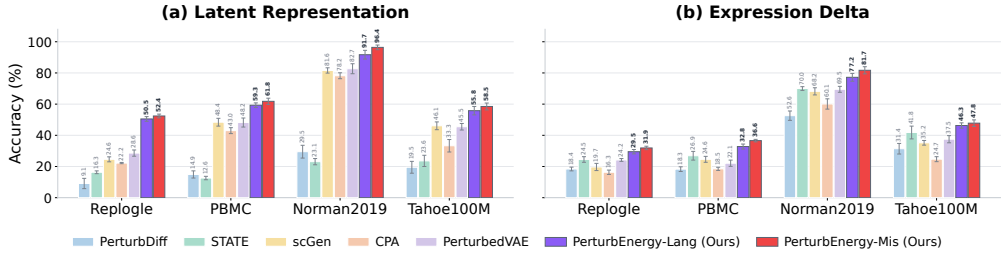

Figure 1: Perturbation classification accuracy. Existing methods show limited perturbation recognition rate both in representation and in expression.

depends on two key factors: (i) whether the perturbation effect is well described within the representation space, and (ii) whether the model is powerful enough to capture such perturbation effects. Existing methods mainly focus on the second factor where they develop sophisticated models and assume that such models can effectively extract perturbation effects from the latent space. However, they pay considerably less attention to the more fundamental question: *how much perturbation-specific information is actually encoded in the learned representations for the model to exploit?* Addressing this issue is non-trivial due to the inherent complexity of cellular systems. Gene expression is influenced by a wide range of context-dependent factors, including cell type, donor background, and culture conditions. Moreover, biological heterogeneity Altschuler & Wu (2010) further complicates the problem: even cells under identical experimental conditions may differ in their physiological states or microenvironments, resulting in substantially different expression profiles. Collectively, all these variations can overwhelm perturbation-induced signals in the representation space, making it difficult for models to distinguish true perturbation effects from spurious and non-generalizable correlations. Although several approaches, such as stVAE Russkikh et al. (2020) and PerturbedVAE Jiang et al. (2026), claim to learn perturbation-specific representations, they remain largely heuristic: These methods typically rely on contrastive or adversarial objectives to implicitly regularize representation learning, providing only soft constraints without a theoretical guarantee that perturbation-specific signals can be well preserved. We conduct an empirical analysis examining such concern, as shown in Table 3 and Figure 1. We find that existing methods, including those specifically designed to learn perturbation representations, exhibit limited perturbation distinctiveness both in the latent space and in their predicted expression profiles. These observations motivate the central question of this work: *how can we directly enrich cellular representations with perturbation-specific information to improve cellular response prediction?*

In this paper, we propose a novel method named **PerturbEnergy**, which leverages an energy-based model to address this issue. The philosophy behind our method is that an energy function provides an effective mechanism for reshaping the distribution landscape Pang & Wu (2021); Zhou et al. (2022). Specifically, to enhance perturbation-specific information in the representation space, we explicitly formulate the latent prior in a normalized energy-based form. The energy model measures the compatibility between latent representations and perturbations, assigning lower energy (higher probability mass) to compatible pairs while assigning higher energy to others. Unlike existing methods that rely on implicit constraints, this design addresses the problem more directly by tilting the latent prior toward perturbation-preferred regions, thereby encouraging the learned representations to contain richer perturbation-specific information. Moreover, the proposed energy model can be easily integrated into autoencoder architectures. We show that the Evidence Lower Bound (ELBO) associated with the energy model can be optimized separately from the remaining architecture while remaining equivalent to the ELBO of the complete autoencoder. Our energy formulation also offers an additional computational advantage: the gradient of the normalized energy function can be derived analytically, while the samples required for optimization can be efficiently obtained through negative sampling or Langevin dynamics. Together, these properties allow the proposed energy model to be optimized independently of specific model architecture. Finally, we theoretically analyze the effectiveness of PerturbEnergy and prove that *it can increase a lower bound on the mutual information between perturbations and latent representations.* To validate our method, we conduct comprehensive experiments on several benchmark datasets. The results show that PerturbEnergy substantially enhances perturbation-specific information, particularly in the latent representation space, with improvements of up to 83.2%. It also consistently achieves state-of-the-art performance on top-50/100/200 differentially expressed gene (DEG) prediction and competitive performance on highly variable gene (HVG) prediction.

2 Related Work

We summarize existing methods for cellular perturbation prediction from a representation learning perspective. Linear models Ahlmann-Eltze et al. (2024) perform regression in the expression space or on representations from foundation models such as scGPT Cui et al. (2024) to predict responses. VAE-based methods model perturbation effects in latent space. For example, scGen Lotfollahi et al. (2019) and Biolord Piran et al. (2024) represent perturbation effects as additive latent factors; SAMS-VAE Bereket & Karaletsos (2023) models perturbations using sparse latent representations. Other approaches design more advanced architectures to capture perturbation effects in the representation space. GEARS Roohani et al. (2024) learns genetic perturbation representations using Graph Neural Networks (GNNs); STATE Adduri et al. (2026) uses a Transformer to model perturbation-induced latent shifts; and diffusion or flow-based methods He et al. (2026); Yuan et al. (2026); Klein et al. (2025); Bunne et al. (2023) learn transitions between control and perturbed representations through diffusion or flow matching process. All these methods focus on learning perturbation effects from the representation space, but ignore whether the latent space contains sufficient perturbation-specific information. Although some studies claim to learn perturbation-awared representations, they remain intuitive: PerturbedVAE Jiang et al. (2026) employs a Contrastive object to disentangle perturbation representations from other factors; stVAE Russkikh et al. (2020) and CPA Lotfollahi et al. (2023) use adversarial training to promote such disentanglement. These auxiliary objectives provide only implicit constraints without a theoretical guarantee that perturbation-specific information is well preserved in the latent space.

3 Method

3.1 Preliminary

Before introducing our method, we first formalize the basic modeling framework and then explain why perturbation-specific information is important. Cellular perturbation prediction aims to predict gene expression induced by genetic or drug perturbations and we study it from a probabilistic perspective, assuming that each response is jointly determined by the cellular context and the applied perturbation. Formally, Let $x \in \mathbb{R}^G$ denotes the gene-expression profile of a perturbed cell, $a \in \mathcal{A}$ denotes its perturbation, and $c \in \mathcal{C}$ denotes its cellular context.[1] Given observations $\mathcal{D} = \{(x_i, c_i, a_i)\}_{i=1}^N$, the goal is to learn a distribution $p(x|c, a)$ that predicts the cellular expression profile. To learn perturbation-specific effect, we decompose the latent representation into two components $z = (z_b, z_a)$, where z_b captures the basal cellular state that is invariant to perturbation and z_a captures perturbation-specific variation. The resulting joint distribution factorizes as:

$$p(x, z_b, z_a|c, a) = p(x|z_b, z_a)p(z_b, z_a|c, a), \tag{1}$$

where $p(x|z_b, z_a)$ is the perturbation expression distribution conditioned on latent representations and can be parameterized by a generative decoder $p_\theta(x|z_b, z_a)$ with parameter θ; $p(z_b, z_a|c, a)$ is the latent prior distribution, we adopt the variational inference like prior work Jiang et al. (2026) to approximate it as:

$$q_\phi(z_b, z_a|x_{ctr}, c, a) = q_\phi(z_b|x_{ctr}, c)q_\phi(z_a|x_{ctr}, a, c), \tag{2}$$

where x_{ctr} is the controlled (unperturbed) cell expression under the same context c. To model such a joint distribution, existing approaches Jiang et al. (2026); Yuan et al. (2026); Adduri et al. (2026) either use VAE to learn perturbation-aware representations via implicit constraints or employ powerful decoders $p_\theta(x|z_b, z_a)$ to learn perturbation effects from the latent space. However, these methods do not directly address the concern: **does the latent space contain sufficient perturbation-specific information for the model to exploit?** Actually, the amount of perturbation-specific information retained in the latent space directly affects how easily the model can learn perturbation effects. More informative representations make generalizable perturbation patterns easier to identify and can therefore improve cellular response prediction. Our preliminary analysis supports this. By comparing CPA and PerturbedVAE in terms of perturbation classification accuracy and root mean square error (RMSE) on differentially expressed genes (DEGs), as shown in Table 3 and Table 2, we observe that richer perturbation-specific information is associated with better perturbation prediction

[1] Here, x denotes the perturbed expression profile and should rigorously be written as x_{pert}. For simplicity, we use x throughout the remainder of the paper unless otherwise specified.

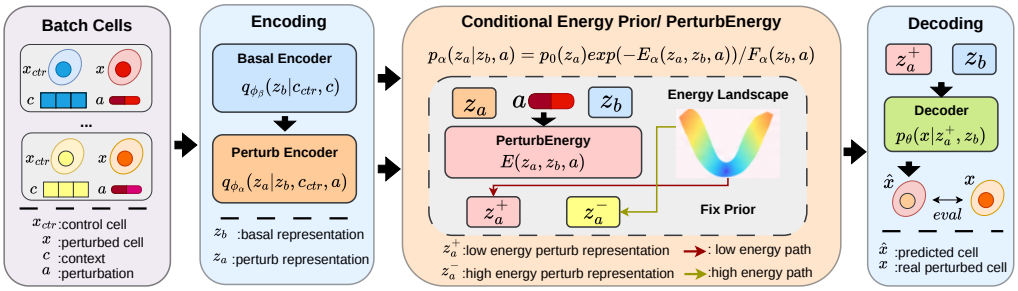

Figure 2: Our PerturbEnergy framework. It reshapes the latent prior to perturbation preferred regions by assigning low energy to perturbation-compatible representations. Thus we can sample the representation with more perturbation-specific information for the decoder.

performance (CPA vs PerturbedVAE; Acc: 22.2% vs. 28.6%; Top50 RMSE: 0.3462 vs. 0.2130). However, we also find existing methods still capture limited perturbation-specific information within their latent space (less than 50% Acc on Replogle, PBMC and Tahoe100M, as shown in Figure 1). These observations motivate the central question of this work: *how can we explicitly enrich latent representations with perturbation-specific information to improve cellular response prediction?*

3.2 Conditional Energy-Based Perturbation Prior and Energy-ELBO

To address the above concern, we propose **PerturbEnergy**, which introduces a conditional energy-based prior over the perturbation latent space. The framework is shown in Figure 2. Our design is motivated by the observation that an energy function can serve as a flexible mechanism for revising a target distribution Pang & Wu (2021); Zhou et al. (2022). Specifically, we factorize the joint latent prior in Eq. equation 1 as:

$$p(z_b, z_a|c, a) = p_\beta(z_b|c)p_\alpha(z_a|z_b, a), \tag{3}$$

where $p_\beta(z_b|c)$ denotes the marginal prior over the basal representation, and $p_\alpha(z_a|z_b, a)$ denotes the conditional prior over the perturbation representation. To enrich the latent space with perturbation-specific information, we formulate $p_\alpha(z_a|z_b, a)$ as a conditional energy-based distribution:

$$p_\alpha(z_a|z_b, a) = p_0(z_a)\exp\left[-E_\alpha(z_a, z_b, a)\right]/F_\alpha(z_b, a), \tag{4}$$

where $p_0(z_a)$ denotes the prior without perturbation constraints (i.e, the prior used in existing works for perturbation space). $E_\alpha(z_a, z_b, a)$ is the energy-based model that revises the prior according to the perturbation: for a given basal state z_b and perturbation a, it assigns lower energy (higher probability mass) to the perturbation representation z_a that are more compatible with this condition and assigns higher energy to others. Unlike existing approaches Yuan et al. (2026); Russkikh et al. (2020) that encourage perturbation-specific information through implicit constraints, our method directly modifies the *latent prior* of the perturbation space, explicitly tilting it toward the perturbation-preferred region. Here, $E_\alpha(z_a, z_b, a)$ can be implemented as a scalar-valued neural network parameterized by α. $F_\alpha(z_b, a)$ is the corresponding partition function ensuring the normalization of the distribution, which is defined as:

$$F_\alpha(z_b, a) = \int p_0(z_a)\exp\left(-E_\alpha(z_a, z_b, a)\right)\mathrm{d}z_a. \tag{5}$$

Based on this formulation, we then consider how to optimize such a model. Our target is to fit the real perturbed cell expression distribution, formally to maximize the $log\, p_\Theta(x|c, a)$. However, given $log\, p_\Theta(x|c, a) = log \int p_\theta(x|z_b, z_a)p(z_b, z_a|c, a)dz_b dz_a$, it has two challenges: (i) the integral over latent space is intractable, and (ii) the joint latent prior $p(z_b, z_a|c, a)$ is difficult to model. Therefore we adopt the variational inference in Eq. equation 2 to estimate the energy-formed prior as:

$$q_\phi(z_b, z_a|x_{ctr}, c, a) = q_{\phi_\beta}(z_b|x_{ctr}, c)q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a), \tag{6}$$

where $q_{\phi_\beta}(z_b|x_{ctr}, c)$ is the basal state encoder estimating the marginal prior $p_\beta(z_b|c)$ and $q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a)$ is the perturbation encoder estimating the conditional energy prior $p_\alpha(z_a|z_b, a)$. This variational formulation leads to the Evidence Lower Bound (ELBO) of $log\, p_\Theta(x|c, a)$ as:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(z_b, z_a|x_{ctr}, c, a)}\left[log\frac{p_\Theta(x, z_b, z_a|c, a)}{q_\phi(z_b, z_a|x_{ctr}, c, a)}\right] \tag{7}$$

The optimization of the ELBO includes the complete model parameters $\{\Theta, \phi\} = \{\theta, \alpha, \phi_\beta, \phi_\alpha\}$, where θ is the parameter of generative decoder, ϕ_β is the parameter of basal state encoder, ϕ_β is the parameter of perturbation encoder, and α is our energy-based model parameter. We focus on the optimization corresponding to energy-based model, then we will prove that this optimization is equivalent to the optimization of entire ELBO. The Kullback-Leibler (KL) divergence between $q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a)$ and $p_\alpha(z_a|z_b, a)$ is derived as:

$$\begin{aligned} D_{\mathrm{KL}}\left(q_{\phi_\alpha}||p_\alpha\right) &= \mathbb{E}_{q_{\phi_\alpha}}\left[log\, q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a) - log\, p_0(z_a) + E_\alpha(z_a, z_b, a) + \log F_\alpha(z_b, a)\right] \\ &= D_{\mathrm{KL}}\left(q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a)||p_0(z_a)\right) + \mathbb{E}_{q_{\phi_\alpha}}\left[E_\alpha(z_a, z_b, a)\right] + \log F_\alpha(z_b, a). \end{aligned} \tag{8}$$

Eq. equation 8 provides an intuitive understanding of PerturbEnergy: the KL divergence is not only bounded by regularization to the conventional prior (i.e., $D_{\mathrm{KL}}\left(q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a)||p_0(z_a)\right)$), it also considers the compatibility between the inferred perturbation representation and the condition (z_b, a). Since $\log F_\alpha(z_b, a)$ is independent of z_a, optimizing Eq. equation 8 encourages $q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a)$ to prefer z_a with more perturbation-specific information. In this paper, we term the **negative KL** $-D_{\mathrm{KL}}\left(q_{\phi_\alpha}||p_\alpha\right)$ in Eq. equation 8 as **Energy-ELBO** and its relationship with the full objective $\mathcal{L}_{\mathrm{ELBO}}$ is formalized in Theorem 1. The detailed proof is provided in Appendix A.1.1.

Theorem 1. *Energy-ELBO $-D_{\mathrm{KL}}\left(q_{\phi_\alpha}||p_\alpha\right)$ is a part of the $\log p_\Theta(x|c, a)$ ELBO, specifically we have:*

$$\mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_\phi}\left[log\, p_\theta(x|z_b, z_a)\right] - D_{\mathrm{KL}}(q_{\phi_\beta}||p_\beta) + \mathbb{E}_{q_{\phi_\beta}}\left[-D_{\mathrm{KL}}\left(q_{\phi_\alpha}||p_\alpha\right)\right]. \tag{9}$$

Therefore, given specific (z_b, a) the optimization of Energy-ELBO is exactly equivalent to optimizing $\mathcal{L}_{\mathrm{ELBO}}$ at the complete model.

3.3 Optimization for PerturbEnergy

Although we have derived an explicit form of the Energy-ELBO, optimizing PerturbEnergy still face a problem that the partition function $F_\alpha(z_b, a)$ in Eq. equation 5 requires integration over the entire perturbation latent space and is generally intractable. Fortunately, its gradient admits a tractable form, which is described in Lemma 1. The proof is provided in Appendix A.1.2.

Lemma 1. *Given the partition function is defined as Eq. equation 5, the gradient with respect to the energy parameters α is:*

$$\nabla_\alpha \log F_\alpha(z_b, a) = -\mathbb{E}_{p_\alpha(z_a|z_b,a)}\left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right]. \tag{10}$$

Therefore, the gradient of $\mathcal{L}_{\mathrm{ELBO}}$ with respect to the energy parameters α is as follows, the derivation is given in Appendix A.1.3.

$$\nabla_\alpha \mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_{\phi_\beta}}\{-\mathbb{E}_{q_{\phi_\alpha}}\left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right] + \mathbb{E}_{p_\alpha}\left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right]\}. \tag{11}$$

Since the perturbation representation is sampled based on the fixed condition (z_b, a), and our objective is to *maximize* the EBLO, we consider the corresponding *negative gradient*:

$$\nabla_\alpha \mathcal{J}_{\mathrm{E}}(\alpha) = \underbrace{\mathbb{E}_{q_{\phi_\alpha}}\left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right]}_{\text{positive phase}} - \underbrace{\mathbb{E}_{p_\alpha}\left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right]}_{\text{negative phase}}. \tag{12}$$

This gradient is exactly the same as maximizing the Energy-ELBO in Eq. equation 8, which again shows the equivalence of optimizing the Energy-ELBO and the complete ELBO. The *positive phase* for this gradient is easy to calculate since positive samples can be obtained from the real cell data via $z_a^+ \sim q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a)$. In contrast, the *negative phase* requires sampling z_a from $p_\alpha(z_a|z_b, a)$, which is intractable as it involves the partition function. We provide two practical solutions. The first is based on physical function of PerturbEnergy, which assigns lower energy to compatible representations and higher energy to incompatible ones. Accordingly, we can sample the *mismatched* perturbation representations for the negative phase, formally via $z_a^- \sim q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a^-)$ where a^- denotes a perturbation different from the true a. The second solution exploits an additional property of our energy-based prior: *its gradient with respect to z_a can be calculated directly*. This enables us to use Langevin dynamics to approximately sample from $p_\alpha(z_a|z_b, a)$ for the negative phase, which is described as follows and the algorithm can be found in Appendix A.2.1.

Langevin Sampling: Since the partition function $F_\alpha(z_b, a)$ is independent of z_a, the gradient of $log\, p_\alpha(z_a|z_b, a)$ with respect to z_a can be written as:

$$\nabla_{z_a} \log\, p_\alpha(z_a|z_b, a) = \nabla_{z_a} \log\, p_0(z_a) - \nabla_{z_a} E_\alpha(z_a, z_b, a), \tag{13}$$

We adopt the Euler-Maruyama discretized Langevin, which updates the sample as: $z_a^{(k+1)} = z_a^{(k)} + \frac{\eta}{2}\nabla_{z_a} \log\, p_\alpha(z_a^{(k)}|z_b, a) + \sqrt{\eta}\epsilon^{(k)}$, where $z_a^{(k)}$ denotes the sample at the k-th step, $\eta > 0$ is the Langevin step size, and $\epsilon^{(k)} \sim \mathcal{N}(0, I)$ is the random noise term corresponding to Brownian motion. Assuming a Gaussian prior $p_0(z_a)$ and combining it with Eq. equation 13, we obtain:

$$z_a^{(k+1)} = z_a^{(k)} - \frac{\eta}{2}\left[\nabla_{z_a} E_\alpha(z_a^{(k)}, z_b, a) + z_a^{(k)}\right] + \sqrt{\eta}\epsilon^{(k)}. \tag{14}$$

After K steps, we use $z_a^{(K)}$ as the negative sample. Intuitively, the positive phase lowers the energy of latent representations inferred from real perturbation and cell observations, whereas the negative phase raises the energy of representations favored by the current model but not supported by the data. In other word, the negative phase corrects representations to which the model has incorrectly assigned low energy. Using either of the two solutions, we can obtain negative samples for optimization. Together with the positive samples, the gradient for optimizing PerturbEnergy can then be calculated via Eq. equation 12.

Optimization: We implement the optimization of the entire model, including the varitional encoder q_{ϕ_α}, q_{ϕ_β}, generator p_θ, and the PerturbEnergy E_α. The target is to *maximize* the ELBO in Eq. equation 9, which can be derived as:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi}\left[log\, p_\theta\right] - D_{\text{KL}}(q_{\phi_\beta}||p_\beta) - \mathbb{E}_{q_{\phi_\beta}}\left[D_{KL}(q_{\phi_\alpha}||p_0)\right] - \mathbb{E}_{q_{\phi_\beta}}\left[\mathbb{E}_{q_{\phi_\alpha}} E_\alpha + logF_\alpha\right]. \tag{15}$$

We use Maximum Mean Discrepancy (MMD) as the surrogate to reconstruction term $\mathbb{E}_{q_\phi}\left[log\, p_\theta\right]$. For the two KL-divergence terms, we assume the Gaussian prior for p_0 and p_β. The gradient of E_α with respect to ϕ_α and ϕ_β is easy to calculate since the gradient $\nabla_{z_a/z_b} E_\alpha$ is direct and such gradient can be backpropagated to ϕ_α/ϕ_β via Jacobian $\frac{\partial z_a}{\partial \phi_\alpha}/\frac{\partial z_b}{\partial \phi_\beta}$. The last one is the gradient of $logF_\alpha$ with respect to ϕ_β. According to Lemma 1, its gradient with respect to z_b takes a similar form: $-\mathbb{E}_{p_\alpha}\left[\nabla_{z_b} E_\alpha\right]$, which can be further backpropagated to ϕ_β via Jacobian $\frac{\partial z_b}{\partial \phi_\beta}$. Together with $\nabla_\alpha \mathcal{J}_{\text{E}}(\alpha)$ in Eq. equation 12, the whole model can be optimized as: Given batch data (x_{ctr}, x) under the same context c with perturbation a, first calculate the basal representations z_b via q_{ϕ_β}, then generate N positive batch samples $\{z_{a,j}^+\}_{j=1}^N$ via q_{ϕ_α} and N negative batch samples $\{z_{a,j}^-\}_{j=1}^N$ via Langevin or Mismatching. Thus α is updated by $\nabla_\alpha \mathcal{J}_{\text{E}}(\alpha)$ whereas $\{\theta, \phi_\beta, \phi_\alpha\}$ are updated by $\{\nabla_\theta \mathcal{L}_{\text{ELBO}}, \nabla_{\phi_\beta} \mathcal{L}_{\text{ELBO}}, \nabla_{\phi_\alpha} \mathcal{L}_{\text{ELBO}}\}$. The Algorithm is provided in Appendix A.2.2

Inference: We adopt the importance sampling in the inference stage. For unperturbed cell (x_{ctr}, c) and perturbation a: (i) calculate the basal state z_b via q_{ϕ_β}; (ii) sample multiple perturbation representations $\{z_{a,i}\}_{i=0}^M$ via q_{ϕ_α}; (iii) select the z_a with the lowest energy E_α; (vi) generate cellular response x via p_θ based on the z_b and z_a. The algorithm is provided in Appendix A.2.3.

3.4 Discussion

We analyze the effectiveness of PerturbEnergy in enhancing perturbation-specific information. Our theoretical finding is formalized as Theorem 2, whose proof is provided in Appendix A.1.4.

Theorem 2. *Optimizing PerturbEnergy leading to the increase of the variational mutual information lower bound between the perturbation and the latent representation. Specifically, the relation is:*

$$I(A; Z_a|Z_b, C) \geq H(A|Z_b, C) + \mathbb{E}\left[\log r_\alpha(A|Z_a, Z_b, C)\right], \tag{16}$$

$$r_\alpha(a|z_a, z_b, c) = \frac{\pi(a|z_b, c)\exp(-E_\alpha(z_a, z_b, a))}{\sum_{a' \in \mathcal{A}} \pi(a'|z_b, c)\exp(-E_\alpha(z_a, z_b, a'))} \tag{17}$$

where $\pi(a|z_b, c)$ is the true conditional prior of perturbation, $I(.;.|Z_b, C)$ denotes the conditional mutual information and $H(.|Z_b, C)$ is the conditional entropy.

4 Experiments

4.1 Experimental Settings

Baselines: We compare with following baselines: (i) Linear methods: we use Linear model on cell expression and scGPT Cui et al. (2024) representations; (ii) VAE-based methods: we use scGen Lotfollahi et al. (2019), CPA Lotfollahi et al. (2023), stVAE Russkikh et al. (2020), and PerturbedVAE Jiang et al. (2026); iii) Other architecture: we use PerturbDiff Yuan et al. (2026), Squidiff He

et al. (2026), CellFlow Klein et al. (2025) and STATE Adduri et al. (2026). The detailed settings of all these baselines are provided in Appendix A.3.2.

Datasets: We use four widely used benchmark datasets: Replogle Replogle et al. (2022), PBMC Yuan et al. (2026), Norman2019 Norman et al. (2019), and Tahoe100M Zhang et al. (2025). For a fair comparison, we follow the standardized experimental settings used in STATE Adduri et al. (2026) and VC benchmark Mao et al. (2026). For all datasets, we use top 2,000 highly variable genes (HVGs). The basic statistics of these datasets are summarized in Table 1, and detailed descriptions are provided in Appendix A.3.1.

Table 1: **Data statistics.** Number of cells(#Cells), perturbations(#Pert.), context(#CT.) and batches(#Batches).

Dataset	#Cells	#Pert.	#CT.	#Batches
Replogle	643.3K	2,023	4	56
PBMC	9.7M	90	18	12
Norman2019	111.4K	236	1	8
Tahoe100M	101.2M	1,137	50	14

Metrics: We evaluate all the methods from two perspectives: (i) how well a model reproduces the highly variable gene (HVG) expression (**HVG group**); (ii) how well a model predict the differential expression gene (DEG) expression influenced by the perturbation. (**DEG group**). For the HVG group, we adopt Cell-Eval Adduri et al. (2026) including: (a) correlation measures: Pearson correlation (PCC), Pearson delta correlation (PDCorr), Coefficient of Determination (R^2), log-fold-change Spearman (LFCSpear), log-fold-change cosine (LogFC_cos), direction agreement (DirAgr) and Perturbation discrimination (PDS-ℓ_1/ℓ_2/cos); (b) DE measures: precison (DEPrec), recall (DERec), Jaccard index (DEOverlap), AUROC and AUPRC between the prediction and real top-50 DEGs; (c) distance measures: Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), Energy Distance (EDist) and Sinkhorn Divergence (Sinkhorn); For the DEG group, we calculate RMSE, MAE, PCC and R^2 over the top-50/100/200 DEGs respectively. The full definitions of all these metrics are given in Appendix A.3.3.

Implementation: PerturbEnergy is model-agnostic and and we implement it with a standard VAE backbone. Both encoders use two-layer MLPs with a latent dimension of 512, while context c and perturbation a are represented by learnable embedding tables. $E_\alpha(z_a, z_b, a)$ is implemented as a simple neural network: we concatenate z_a and z_b and feed it to a two-layer MLP and then output the cosine similarity score with the embedding of a. A two-layer MLP decoder predicts the expression of 2,000 HVGs from the concatenated latent representations. We train with Adam Kingma & Ba (2014) at a learning rate of 10^{-3}, using batch sizes and maximum training steps of (256, 20,000) for Norman2019, (512, 40,000) for Replogle, (512, 60,000) for PBMC and (512, 160,000) Tahoe100M. By default, PerturbEnergy-Langevin(Lang) uses $K = 20$ and $\eta = 10^{-4}$, while PerturbEnergy-Mismatch(Mis) randomly samples embeddings from other perturbations as negatives. During inference, we set the candidate number $M = 8$ by default. We apply the same checkpoint selection strategy to all methods, as detailed in Appendix A.3.2.

4.2 Experimental Results and Discussions

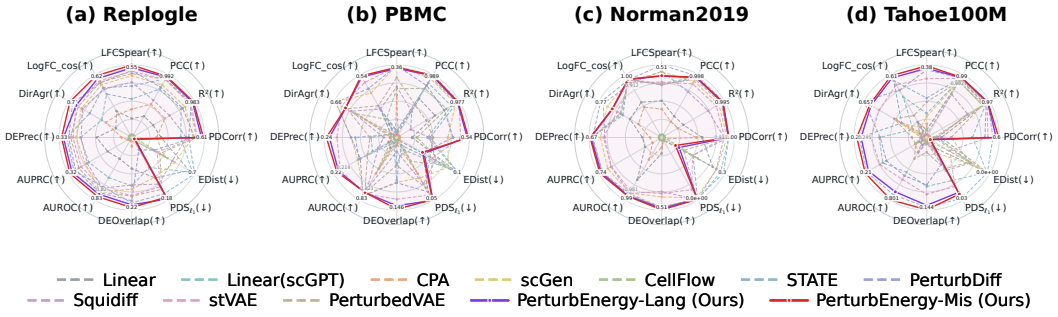

Figure 3: Performance on parts of the HVG group metrics. ↑ / ↓ means the higher/lower is better.

HVG Group. Figure 3 summarizes the performance of all methods on the HVG group, with detailed results provided in Appendix A.3.8. For correlation- and distance-based metrics, including PDCorr, LFCSpear, R^2, and EDist, PerturbEnergy generally achieves competitive performance across all four datasets, demonstrating its ability to capture overall expression patterns and distributional similarities. Moreover, our method consistently outperforms the baselines on differential expression (DE)

Table 2: RMSE (↓) over the true top-K DEGs. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	Replogle			Tahoe100M		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
Linear	0.2738 ±0.0086	0.2390 ±0.0072	0.2045 ±0.0058	0.4536 ±0.0072	0.3866 ±0.0053	0.3032 ±0.0040
Linear (scGPT)	0.2449 ±0.0081	0.2081 ±0.0066	0.1738 ±0.0052	0.3748 ±0.0060	0.3257 ±0.0044	0.2588 ±0.0034
CellFlow	0.2174 ±0.0067	0.1850 ±0.0055	0.1536 ±0.0043	0.3860 ±0.0062	0.3318 ±0.0044	0.2617 ±0.0034
Squidiff	0.2173 ±0.0055	0.1907 ±0.0044	0.1658 ±0.0035	0.4192 ±0.0071	0.3889 ±0.0051	0.3231 ±0.0039
PerturbDiff	0.2612 ±0.0096	0.2198 ±0.0079	0.1808 ±0.0063	0.3558 ±0.0070	0.3079 ±0.0051	0.2462 ±0.0040
STATE	0.2024 ±0.0064	0.1727 ±0.0052	0.1447 ±0.0041	0.3356 ±0.0041	0.2935 ±0.0030	0.2342 ±0.0023
scGen	0.2336 ±0.0078	0.1992 ±0.0065	0.1661 ±0.0051	0.4462 ±0.0081	0.3818 ±0.0059	0.3018 ±0.0046
CPA	0.3462 ±0.0132	0.2905 ±0.0110	0.2363 ±0.0088	0.7070 ±0.0103	0.6457 ±0.0075	0.5304 ±0.0056
stVAE	0.2147 ±0.0071	0.1795 ±0.0057	0.1484 ±0.0044	0.3293 ±0.0057	0.2883 ±0.0041	0.2315 ±0.0031
PerturbedVAE	0.2130 ±0.0071	0.1817 ±0.0058	0.1514 ±0.0047	0.3984 ±0.0075	0.3476 ±0.0055	0.2791 ±0.0042
PerturbEnergy-Lang	0.1870 ±0.0061	0.1591 ±0.0049	0.1335 ±0.0039	**0.3114** ±0.0051	**0.2719** ±0.0037	**0.2179** ±0.0029
PerturbEnergy-Mis	**0.1780** ±0.0057	**0.1522** ±0.0046	**0.1285** ±0.0036	0.3123 ±0.0051	0.2723 ±0.0037	0.2179 ±0.0029

Method	PBMC			Norman2019		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
Linear	0.1065 ±0.0046	0.0901 ±0.0037	0.0740 ±0.0029	0.4105 ±0.0264	0.3136 ±0.0208	0.2313 ±0.0155
Linear (scGPT)	0.0979 ±0.0045	0.0816 ±0.0036	0.0656 ±0.0029	0.2810 ±0.0090	0.2422 ±0.0073	0.1837 ±0.0055
CellFlow	0.0821 ±0.0024	0.0678 ±0.0019	0.0541 ±0.0015	0.1603 ±0.0145	0.1292 ±0.0118	0.1007 ±0.0089
Squidiff	0.1271 ±0.0051	0.1106 ±0.0042	0.0931 ±0.0033	0.2041 ±0.0106	0.1743 ±0.0084	0.1447 ±0.0068
PerturbDiff	0.0829 ±0.0031	0.0718 ±0.0025	0.0602 ±0.0020	**0.1248** ±0.0088	**0.1035** ±0.0067	**0.0836** ±0.0051
STATE	0.0935 ±0.0033	0.0774 ±0.0027	0.0619 ±0.0022	0.1431 ±0.0117	0.1168 ±0.0091	0.0922 ±0.0065
scGen	0.0960 ±0.0049	0.0787 ±0.0040	0.0625 ±0.0031	0.1729 ±0.0102	0.1404 ±0.0082	0.1108 ±0.0065
CPA	0.1581 ±0.0069	0.1317 ±0.0054	0.1062 ±0.0041	0.4419 ±0.0282	0.3370 ±0.0219	0.2477 ±0.0162
stVAE	0.0729 ±0.0022	0.0613 ±0.0018	0.0500 ±0.0014	0.2110 ±0.0146	0.1656 ±0.0114	0.1260 ±0.0084
PerturbedVAE	0.1071 ±0.0032	0.0925 ±0.0025	0.0777 ±0.0020	0.1400 ±0.0123	0.1186 ±0.0100	0.0953 ±0.0075
PerturbEnergy-Lang	**0.0637** ±0.0020	**0.0542** ±0.0016	**0.0448** ±0.0013	0.1434 ±0.0113	0.1176 ±0.0086	0.0927 ±0.0062
PerturbEnergy-Mis	0.0654 ±0.0020	0.0557 ±0.0016	0.0462 ±0.0013	0.1376 ±0.0110	0.1135 ±0.0083	0.0898 ±0.0061

metrics, including DEPrec, AUROC, AUPRC, and DEOverlap. This improvement highlights the effectiveness of explicitly modeling perturbation-specific information, which is particularly important for identifying differentially expressed genes (DEGs) associated with perturbations.

DEG Group. Unlike the HVG group which evaluates general gene expression patterns, the DEG group metrics focus on predicting perturbation-specific responses. We conduct experiments for the DEG group metrics and report the RMSE results in Table 2, with the remaining metrics and corresponding analysis provided in the Appendix A.3.7. We have the following key observations: (i) Although linear methods can capture general expression patterns in the HVG group, they struggle to predict perturbation-induced changes, suggesting that simple regression models may focus on house keeping gene patterns rather than perturbation effects. (ii) CellFlow, Squidiff, and PerturbDiff model perturbation effects through diffusion or flow-matching processes, but their performance varies across datasets. For example, PerturbDiff achieves state-of-the-art performance on Norman2019 but struggles on Replogle. This variation suggests that relying primarily on model expressiveness may not consistently capture perturbation effects when they are obscured by other sources of cellular variation. Notably, STATE achieves competitive performance, potentially benefiting from batch-level self-attention that mitigates cellular heterogeneity and its large model capacity. (iii) Among representation learning-based methods, PerturbedVAE achieves better performance than other baselines, highlighting the potential benefits of explicitly modeling perturbation-specific information. (iv) Despite using a simple MLP backbone, PerturbEnergy achieves the best performance on Replogle, PBMC, and Tahoe100M and the second-best performance on Norman2019. These results demonstrate the effectiveness of our design in facilitate capturing perturbation-specific effects.

Perturbation Study. To evaluate how effectively different methods capture perturbation-specific information, we conduct an analysis using two types of inputs: (i) latent representations and (ii) expression differences between predicted perturbed cells and observed control cells. For each setting, we train a two-layer MLP classifier to predict perturbation labels. The results are reported in Table 3 and Figure 1. For latent representations, scGPT captures limited perturbation effect as it's likely to be subtilized; STATE and PerturbDiff achieve limited classification accuracy, as their representations primarily capture basal cellular states; VAE-based methods perform better by incorporating pertur-

Table 3: Perturbation classification accuracy. Best in **bold**, second best underlined.

Method	Latent Representation				Expression Delta			
	Replogle	PBMC	Norman2019	Tahoe100M	Replogle	PBMC	Norman2019	Tahoe100M
scGPT	18.5±1.1	33.1±1.0	74.4±2.9	38.2±1.9	21.5±2.5	24.8±1.4	73.1±2.7	39.9±2.3
PerturbDiff	9.1±3.3	14.9±2.3	29.5±4.1	19.5±3.8	18.4±1.2	18.3±1.5	52.6±3.0	31.4±3.4
STATE	16.3±0.9	12.6±1.1	23.1±2.0	23.6±3.6	24.5±1.8	26.9±2.8	70.0±1.2	41.8±4.1
scGen	24.6±1.6	48.4±2.5	81.6±1.7	46.1±2.5	19.7±2.3	24.6±1.9	68.2±2.3	35.2±1.5
CPA	22.2±0.4	43.0±1.9	78.2±1.9	33.3±4.0	16.3±1.4	18.5±1.0	60.1±3.3	24.7±1.6
PerturbedVAE	28.6±2.0	48.2±2.9	82.7±3.2	45.5±2.1	24.2±0.9	22.1±2.0	69.5±1.9	37.5±2.3
PerturbEnergy-Lang	50.5±1.5	59.3±1.5	91.7±2.8	55.8±2.7	29.5±1.4	32.8±1.6	77.2±2.5	46.3±1.7
PerturbEnergy-Mis	**52.4±1.2**	**61.8±2.0**	**96.4±1.5**	**58.5±2.2**	**31.9±1.1**	**36.6±0.8**	**81.7±2.3**	**47.8±2.1**

bation information into the latent space. Besides, richer perturbation-specific information generally leads to better perturbation prediction performance. For example, PerturbedVAE achieves higher accuracy than CPA and outperforms CPA in Top-50/100/200 RMSE. (See Table 2). However, their accuracy remains below 50% on Replogle, PBMC, and Tahoe100M, suggesting that existing representation learning strategies may not sufficiently preserve perturbation-specific information. For expression differences, STATE achieves competitive performance, whereas PerturbDiff struggles compared with other baselines. This variation suggests that relying primarily on model may not consistently capture perturbation effects. VAE-based methods also exhibit degraded performance, potentially because cellular complexity obscures perturbation-induced changes in the expression. In contrast, PerturbEnergy achieves the best performance in both settings, with particularly substantial improvements in latent representation classification (83.2% on Replogle and 28.2% on PBMC). These results demonstrate the effectiveness of our method in enriching latent representations with perturbation-specific information.

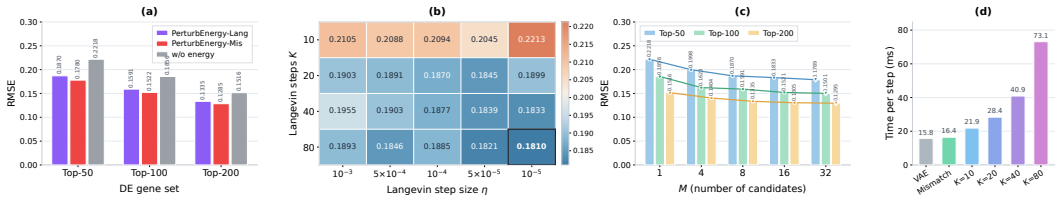

Figure 4: Ablation and other study on Replogle. (a) is the ablation; (b) is the hyperparameter study about Langevin steps/size; (c) is the influence of candidate number M. (d) is the training efficiency.

Ablation and Other Study. We conduct ablation, hyperparameter and efficiency studies on Replogle, as shown in Figure 4. Removing the energy function consistently degrades performance, where the model reduces to a standard VAE without explicitly modeling perturbation-specific information. We then examine the effects of Langevin steps K and step size η using RMSE on the top-50 DEGs. Larger K and smaller η generally improve performance, whereas insufficient Langevin steps lead to degraded results. We also find that Mismatch generally has the same training efficiency as VAE (16.4 vs 15.6 ms/step) while Langevin with larger K significantly decreases the training speed. As a balance, we set $K = 20$ and $\eta = 10^{-4}$ by default, which generally make the sample region with a flat gradient. Finally, we examine the effect of the candidate number M during inference. Performance deteriorates with too few candidates ($M \leq 4$), while increasing M to 16 or 32 yields only marginal improvements. We therefore set $M = 8$ as a balance.

Other Discussion. Due to space limitations, we provide an analysis of the training scale/model dimension in the Appendix A.3.5. We also demonstrate the compatibility of PerturbEnergy with different VAE backbones in Appendix A.3.4. The visualizations are provided in Appendix A.3.6.

5 Conclusion

In this paper, we introduce PerturbEnergy, a framework designed to enhance perturbation-specific information for cellular perturbation prediction. PerturbEnergy uses an energy function to reshape the latent prior, directing it toward perturbation preferred regions. We theoretically demonstrate that PerturbEnergy increases the mutual information between the representations and perturbations. Extensive experiments on four datasets validate its effectiveness.

6 AI USE STATEMENT

In this work, we used generative AI tools only for polishing the paper writing, including fixing grammar mistakes, making expressions more easy to understand. We have not used generative AI tools for developing our research idea and experiments. The applicable and not applicable details are as follows:

Applicable AI usage for this work: (i) Grammar, spelling mistake finding and fixing; (ii) Expression and writing polishing, AI is used to paraphrase some of my sentences to make it clear and easy to understand, while exactly keeping the main content of the original sentences. (iii) Give suggestions on the paper formatting, including the position of our experimental Tables, Figures.

Not Applicable AI usage for this work: We do not use AI for our research idea developing, including: (i) AI is not used for reviewing related works; (ii) AI is not used for developing methodology and theory of this work; (iii) AI is not used for collecting experimental data; (iv) AI is not used for implementing our method codes;

7 ETHICS STATEMENT

Our work is focused on AI-based prediction of biological cellular perturbation response, aiming to enhance the accuracy of cell prediction for common perturbations. This work is the application of the universal AI methodology in the context of cellular prediction, aiming to solve the practical problems in this scenario. The content of this research work does not involve any ethical issues, there is no risk data release, no harmful content, and no conflicts of interest. However, it should be noted that this method is used to improve the computational prediction accuracy of cell perturbations. Therefore, this work needs to be properly applied and should not be used to assist in harmful biomedical research.

8 REPRODUCIBILITY STATEMENT

We declare that all the work presented in this paper is reproducible. Firstly, in terms of methodology, we have provided rigorous theoretical proofs for each theorem/lemma. Secondly, in terms of the software code, we have made the relevant code of our work public via anonymous link. Furthermore, we adopt the publicly available datasets, data processing methods, and evaluation metrics. This means that the experimental and theoretical results of this work can all be reproduced.

REFERENCES

Abhinav K Adduri, Dhruv Gautam, Beatrice Bevilacqua, Mohsen Naghipourfar, Alishba Imran, Rohan Shah, Noam Teyssier, Rishi Verma, Christopher Carpenter, Basak Eraslan, et al. Predicting cellular responses to perturbation across diverse contexts with state. *Cell*, 2026.

Constantin Ahlmann-Eltze, Wolfgang Huber, and Simon Anders. Deep learning-based predictions of gene perturbation effects do not yet outperform simple linear baselines. *BioRxiv*, pp. 2024–09, 2024.

Steven J Altschuler and Lani F Wu. Cellular heterogeneity: do differences make a difference? *Cell*, 141(4):559–563, 2010.

Michael Bereket and Theofanis Karaletsos. Modelling cellular perturbations with the sparse additive mechanism shift variational autoencoder. *Advances in Neural Information Processing Systems*, 36:1–12, 2023.

Charlotte Bunne, Stefan G Stark, Gabriele Gut, Jacobo Sarabia Del Castillo, Mitch Levesque, Kjong-Van Lehmann, Lucas Pelkmans, Andreas Krause, and Gunnar Rätsch. Learning single-cell perturbation responses using neural optimal transport. *Nature methods*, 20(11):1759–1768, 2023.

Charlotte Bunne, Yusuf Roohani, Yanay Rosen, Ankit Gupta, Xikun Zhang, Marcel Roed, Theo Alexandrov, Mohammed AlQuraishi, Patricia Brennan, Daniel B Burkhardt, et al. How to build

the virtual cell with artificial intelligence: Priorities and opportunities. *Cell*, 187(25):7045–7063, 2024.

Haotian Cui, Chloe Wang, Hassaan Maan, Kuan Pang, Fengning Luo, Nan Duan, and Bo Wang. scgpt: toward building a foundation model for single-cell multi-omics using generative ai. *Nature methods*, 21(8):1470–1480, 2024.

Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7): 1853–1866, 2016.

Luke A Gilbert, Matthew H Larson, Leonardo Morsut, Zairan Liu, Gloria A Brar, Sandra E Torres, Noam Stern-Ginossar, Onn Brandman, Evan H Whitehead, Jennifer A Doudna, et al. Crispr-mediated modular rna-guided regulation of transcription in eukaryotes. *Cell*, 154(2):442–451, 2013.

Luke A Gilbert, Max A Horlbeck, Britt Adamson, Jacqueline E Villalta, Yuwen Chen, Evan H Whitehead, Carla Guimaraes, Barbara Panning, Hidde L Ploegh, Michael C Bassik, et al. Genome-scale crispr-mediated control of gene repression and activation. *Cell*, 159(3):647–661, 2014.

Minsheng Hao, Jing Gong, Xin Zeng, Chiming Liu, Yucheng Guo, Xingyi Cheng, Taifeng Wang, Jianzhu Ma, Xuegong Zhang, and Le Song. Large-scale foundation model on single-cell transcriptomics. *Nature methods*, 21(8):1481–1491, 2024.

Siyu He, Yuefei Zhu, Daniel Naveed Tavakol, Haotian Ye, Yeh-Hsing Lao, Zixian Zhu, Cong Xu, Shradha Chauhan, Guy Garty, Raju Tomer, et al. Squidiff: predicting cellular development and responses to perturbations using a diffusion model. *Nature methods*, 23(1):65–77, 2026.

Wenkang Jiang, Yuhang Liu, Yichao Cai, Erdun Gao, Jiayi Dong, Ehsan Abbasnejad, Lina Yao, and Javen Qinfeng Shi. What makes a representation good for single-cell perturbation prediction? *arXiv preprint arXiv:2605.19343*, 2026.

Martin Jinek, Krzysztof Chylinski, Ines Fonfara, Michael Hauer, Jennifer A Doudna, and Emmanuelle Charpentier. A programmable dual-rna–guided dna endonuclease in adaptive bacterial immunity. *science*, 337(6096):816–821, 2012.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Dominik Klein, Jonas Simon Fleck, Daniil Bobrovskiy, Lea Zimmermann, Sören Becker, Alessandro Palma, Leander Dony, Alejandro Tejada-Lapuerta, Guillaume Huguet, Hsiu-Chuan Lin, et al. Cellflow enables generative single-cell phenotype modeling with flow matching. *bioRxiv*, pp. 2025–04, 2025.

Mohammad Lotfollahi, F Alexander Wolf, and Fabian J Theis. scgen predicts single-cell perturbation responses. *Nature methods*, 16(8):715–721, 2019.

Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Leon Hetzel, Yuge Ji, Ignacio L Ibarra, Sanjay R Srivatsan, Mohsen Naghipourfar, Riza M Daza, Beth Martin, et al. Predicting cellular responses to complex perturbations in high-throughput screens. *Molecular systems biology*, 19(6):MSB202211517, 2023.

Xinjie Mao, Songming Zhang, Qianhong Wen, Xiangyu Wen, Kedu Jin, Hao Wu, Shuizhou Chen, Yuqiang Li, Lei Bai, Qi Liu, et al. Benchmarking virtual cell models for in-the-wild perturbation response. *arXiv preprint arXiv:2604.27646*, 2026.

Ajay Nadig, Joseph M Replogle, Angela N Pogson, Mukundh Murthy, Steven A McCarroll, Jonathan S Weissman, Elise B Robinson, and Luke J O'Connor. Transcriptome-wide analysis of differential expression in perturbation atlases. *Nature Genetics*, 57(5):1228–1237, 2025.

Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

Bo Pang and Ying Nian Wu. Latent space energy-based model of symbol-vector coupling for text generation and classification. In *International Conference on Machine Learning*, pp. 8359–8370. PMLR, 2021.

Zoe Piran, Niv Cohen, Yedid Hoshen, and Mor Nitzan. Disentanglement of single-cell data with biolord. *Nature Biotechnology*, 42(11):1678–1683, 2024.

CZI Cell Science Program, Shibla Abdulla, Brian Aevermann, Pedro Assis, Seve Badajoz, Sidney M Bell, Emanuele Bezzi, Batuhan Cakir, Jim Chaffer, Signe Chambers, et al. Cz cellxgene discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic acids research*, 53(D1):D886–D900, 2025.

Joseph M Replogle, Thomas M Norman, Albert Xu, Jeffrey A Hussmann, Jin Chen, J Zachery Cogan, Elliott J Meer, Jessica M Terry, Daniel P Riordan, Niranjan Srinivas, et al. Combinatorial single-cell crispr screens by direct guide rna capture and targeted sequencing. *Nature biotechnology*, 38(8):954–961, 2020.

Joseph M Replogle, Reuben A Saunders, Angela N Pogson, Jeffrey A Hussmann, Alexander Lenail, Alina Guna, Lauren Mascibroda, Eric J Wagner, Karen Adelman, Gila Lithwick-Yanai, et al. Mapping information-rich genotype-phenotype landscapes with genome-scale perturb-seq. *Cell*, 185(14):2559–2575, 2022.

Diana C Resasco, Fei Gao, Frank Morgan, Igor L Novak, James C Schaff, and Boris M Slepchenko. Virtual cell: computational tools for modeling in cell biology. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 4(2):129–140, 2012.

Yusuf Roohani, Kexin Huang, and Jure Leskovec. Predicting transcriptional outcomes of novel multigene perturbations with gears. *Nature biotechnology*, 42(6):927–935, 2024.

Nikolai Russkikh, Denis Antonets, Dmitry Shtokalo, Alexander Makarov, Yuri Vyatkin, Alexey Zakharov, and Evgeny Terentyev. Style transfer with variational autoencoders is a promising approach to rna-seq data harmonization and analysis. *Bioinformatics*, 36(20):5076–5085, 2020.

Gefei Wang, Tianyu Liu, Jia Zhao, Youshu Cheng, and Hongyu Zhao. Modeling and predicting single-cell multi-gene perturbation responses with sclambda. *bioRxiv*, 2024.

F Alexander Wolf, Philipp Angerer, and Fabian J Theis. Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19(1):15, 2018.

Xinyu Yuan, Xixian Liu, Ya Shi Zhang, Zuobai Zhang, Hongyu Guo, and Jian Tang. Perturbdiff: Functional diffusion for single-cell perturbation modeling. *arXiv preprint arXiv:2602.19685*, 2026.

Jesse Zhang, Airol A Ubas, Richard De Borja, Valentine Svensson, Nicole Thomas, Neha Thakar, Ian Lai, Aidan Winters, Umair Khan, Matthew G Jones, et al. Tahoe-100m: A giga-scale single-cell perturbation atlas for context-dependent gene function and cellular modeling. *BioRxiv*, pp. 2025–02, 2025.

Xiao Zhou, Yong Lin, Renjie Pi, Weizhong Zhang, Renzhe Xu, Peng Cui, and Tong Zhang. Model agnostic sample reweighting for out-of-distribution learning. In *International conference on machine learning*, pp. 27203–27221. PMLR, 2022.

A APPENDIX

APPENDIX CONTENTS

A.1 Theoretical supports 14
A.1.1 Proof of Theorem 1: Equivalence of the Energy-ELBO 14
A.1.2 Proof of Lemma 1: Gradient of the partition function 15
A.1.3 Derivation of the gradient to energy parameters 15
A.1.4 Proof of Theorem 2: Conditional Mutual-Information Lower Bound 16
A.2 Algorithm Description 16
A.2.1 Langevin Sampling Algorithm 16
A.2.2 Optimization Algorithm 16
A.2.3 Inference Algorithm 17
A.3 Experimental Appendix 17
A.3.1 Dataset Descriptions and Settings 17
A.3.2 Baseline Settings 19
A.3.3 Metrics 21
A.3.4 Other VAE Backbone Integration 23
A.3.5 Influence of Training Scale and Model Dimension 24
A.3.6 Visualization 25
A.3.7 Other Metrics on DEG Group 25
A.3.8 Results on HVG Group 25

A.1 THEORETICAL SUPPORTS

A.1.1 PROOF OF THEOREM 1: EQUIVALENCE OF THE ENERGY-ELBO

Here we give a formal proof to the Theorem 1 as follows:

Proof. We begin from with the ELBO of $log\, p_\Theta(x|c,a)$ which can be derived as:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} &= \mathbb{E}_{q_\phi(z_b,z_a|x_{ctr},c,a)}\Big[\log p_\theta(x|z_b,z_a) + \log p(z_b,z_a|c,a) - log q_\phi(z_b,z_a|x_{ctr},c,a)\Big] \\ &= \mathbb{E}_{q_\phi}\left[\log p_\theta(x|z_b,z_a)\right] - D_{KL}(q_\phi(z_b,z_a|x_{ctr},c,a)||p(z_b,z_a|c,a)).\end{aligned} \tag{18}$$

According to our joint prior assumption in Eq. equation 3 and the variational inference formation in Eq. equation 6, we have:

$$p(z_b,z_a|c,a) = p_\beta(z_b|c)p_\alpha(z_a|z_b,a), \tag{19}$$

and:

$$q_\phi(z_b,z_a|x_{ctr},c,a) = q_{\phi_\beta}(z_b|x_{ctr},c)q_{\phi_\alpha}(z_a|z_b,x_{ctr},a). \tag{20}$$

Besides, according to the chain rule of Kullback-Leibler (KL) divergence, where:

$$D_{KL}(P(X,Y)||Q(X,Y)) = D_{KL}(P(X)||Q(X)) + D_{KL}(P(Y|X)||Q(Y|X)). \tag{21}$$

The $\mathcal{L}_{\text{ELBO}}$ then becomes:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} =& \mathbb{E}_{q_\phi}\left[\log p_\theta(x|z_b,z_a)\right] \\ &+ \mathbb{E}_{q_{\phi_\beta}}\left[\log p_\beta(z_b|c) - \log q_{\phi_\beta}(z_b|x_{ctr},c)\right] \\ &+ \mathbb{E}_{q_{\phi_\beta}}\{\mathbb{E}_{q_{\phi_\alpha}}\left[\log p_\alpha(z_a|z_b,a) - \log q_{\phi_\alpha}(z_a|z_b,x_{ctr},a)\right]\}.\end{aligned} \tag{22}$$

The second term $\mathbb{E}_{q_{\phi_\beta}}\left[\log p_\beta(z_b|c) - \log q_{\phi_\beta}(z_b|x_{ctr},c)\right]$ is:

$$-D_{\text{KL}}\left(q_{\phi_\beta}||p_\beta\right). \tag{23}$$

For the $\mathbb{E}_{q_{\phi_\alpha}}\left[\log p_\alpha(z_a|z_b,a) - \log q_{\phi_\alpha}(z_a|z_b,x_{ctr},a)\right]$ in the third term, it actually is:

$$-D_{\text{KL}}\left(q_{\phi_\alpha}||p_\alpha\right). \tag{24}$$

Thus, the complete ELBO is:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi}\left[log\, p_\theta(x|z_b,z_a)\right] - D_{\text{KL}}(q_{\phi_\beta}||p_\beta) + \mathbb{E}_{q_{\phi_\beta}}\left[-D_{\text{KL}}\left(q_{\phi_\alpha}||p_\alpha\right)\right]. \tag{25}$$

Until now, we have the proved the formation in Theorem 1.

In addition, given the definition of our conditional EBM prior in Eq. 4, the complete ELBO can be further derived as:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} =& \mathbb{E}_{q_\phi}\left[log\, p_\theta(x|z_b,z_a)\right] - D_{\text{KL}}(q_{\phi_\beta}||p_\beta) \\ &- \mathbb{E}_{q_{\phi_\beta}}\big\{\mathbb{E}_{q_{\phi_\alpha}}[\log q_{\phi_\alpha}(z_a|z_b,x_{ctr},a) - \log p_0(z_a) + E_\alpha(z_a,z_b,a) + \log F_\alpha(z_b,a)]\big\}.\end{aligned} \tag{26}$$

Since $\log F_\alpha(z_b,a)$ is independent of z_a for fixed (z_b,a), thus:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} =& \mathbb{E}_{q_\phi}\left[log\, p_\theta(x|z_b,z_a)\right] - D_{\text{KL}}(q_{\phi_\beta}||p_\beta) \\ &- \mathbb{E}_{q_{\phi_\beta}}\big\{D_{\text{KL}}\left(q_{\phi_\alpha}(z_a|z_b,x_{ctr},a)||p_0(z_a)\right) + \mathbb{E}_{q_{\phi_\alpha}}\left[E_\alpha(z_a,z_b,a)\right] + \log F_\alpha(z_b,a)\big\},\end{aligned} \tag{27}$$

which gives the exact form as Energy-ELBO described in Eq. equation 8. Therefore, we fully prove Theorem 1. □

A.1.2 Proof of Lemma 1: Gradient of the partition function

Here we give a proof to Lemma 1 as follows:

Proof.

$$\nabla_\alpha \log F_\alpha(z_b, a) = \frac{1}{F_\alpha(z_b, a)} \nabla_\alpha F_\alpha(z_b, a) \tag{28}$$

According to the definition in Eq. equation 5, the partition function is:

$$F_\alpha(z_b, a) = \int p_0(z_a) \exp\left(-E_\alpha(z_a, z_b, a)\right) \mathrm{d}z_a. \tag{29}$$

Thus, we have:

$$\begin{aligned}
\nabla_\alpha \log F_\alpha(z_b, a) &= \frac{1}{F_\alpha(z_b, a)} \nabla_\alpha \int p_0(z_a) \exp\left(-E_\alpha(z_a, z_b, a)\right) \mathrm{d}z_a \\
&= \frac{1}{F_\alpha(z_b, a)} \int p_0(z_a) \nabla_\alpha \exp\left(-E_\alpha(z_a, z_b, a)\right) \mathrm{d}z_a \\
&= \frac{1}{F_\alpha(z_b, a)} \int p_0(z_a) \exp\left(-E_\alpha(z_a, z_b, a)\right) [-\nabla_\alpha E_\alpha(z_a, z_b, a)] \mathrm{d}z_a \\
&= \int \frac{p_0(z_a) \exp\left(-E_\alpha(z_a, z_b, a)\right)}{F_\alpha(z_b, a)} [-\nabla_\alpha E_\alpha(z_a, z_b, a)] \mathrm{d}z_a.
\end{aligned} \tag{30}$$

According to the our conditional energy form described in Eq. equation 4, which is:

$$p_\alpha(z_a | z_b, a) = \frac{p_0(z_a) \exp\left[-E_\alpha(z_a, z_b, a)\right]}{F_\alpha(z_b, a)}. \tag{31}$$

Therefore, we have:

$$\begin{aligned}
\nabla_\alpha \log F_\alpha(z_b, a) &= \int p_\alpha(z_a | z_b, a) [-\nabla_\alpha E_\alpha(z_a, z_b, a)] \mathrm{d}z_a \\
&= -\mathbb{E}_{p_\alpha(z_a | z_b, a)} \left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right].
\end{aligned} \tag{32}$$

Therefore, we finish the proof of Lemma 1. □

A.1.3 Derivation of the gradient to energy parameters

Here we derive the gradient of the entire ELBO corresponding to energy parameters.

Proof. According to the formation of $log\, p_\Theta(x|c, a)$ ELBO described in Eq. equation 9, we have:

$$\begin{aligned}
\nabla_\alpha \mathcal{L}_{\mathrm{ELBO}} &= \nabla_\alpha \mathbb{E}_{q_\phi} \left[log\, p_\theta(x | z_b, z_a)\right] - \nabla_\alpha D_{\mathrm{KL}}(q_{\phi_\beta} || p_\beta) + \nabla_\alpha \mathbb{E}_{q_{\phi_\beta}} \left[-D_{\mathrm{KL}}\left(q_{\phi_\alpha} || p_\alpha\right)\right] \\
&= \mathbb{E}_{q_{\phi_\beta}} \left[-\nabla_\alpha D_{\mathrm{KL}}\left(q_{\phi_\alpha} || p_\alpha\right)\right].
\end{aligned} \tag{33}$$

According to the Energy-ELBO definition in Eq. equation 8, we then have:

$$\begin{aligned}
\nabla_\alpha \mathcal{L}_{\mathrm{ELBO}} &= \mathbb{E}_{q_{\phi_\beta}} \left[-\nabla_\alpha D_{\mathrm{KL}}\left(q_{\phi_\alpha} || p_\alpha\right)\right] \\
&= \mathbb{E}_{q_{\phi_\beta}} \{-\nabla_\alpha D_{\mathrm{KL}}\left(q_{\phi_\alpha}(z_a | z_b, x_{ctr}, a) || p_0(z_a)\right) - \mathbb{E}_{q_{\phi_\alpha}} \left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right] - \nabla_\alpha \log F_\alpha(z_b, a)\}.
\end{aligned} \tag{34}$$

Since $D_{\mathrm{KL}}\left(q_{\phi_\alpha}(z_a | z_b, x_{ctr}, a) || p_0(z_a)\right)$ is unrelated to α, together with Lemma 1, where:

$$\nabla_\alpha \log F_\alpha(z_b, a) = -\mathbb{E}_{p_\alpha(z_a | z_b, a)} \left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right], \tag{35}$$

we have:

$$\nabla_\alpha \mathcal{L}_{\mathrm{ELBO}} = \mathbb{E}_{q_{\phi_\beta}} \{-\mathbb{E}_{q_{\phi_\alpha}} \left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right] + \mathbb{E}_{p_\alpha} \left[\nabla_\alpha E_\alpha(z_a, z_b, a)\right]\}. \tag{36}$$

Therefore, we finish the derivation of the gradient corresponding to energy parameters α. □

A.1.4 Proof of Theorem 2: Conditional Mutual-Information Lower Bound

Here we give a proof to Theorem 2 as follows:

Proof. According to our definition of conditional energy-based prior in Eq. equation 4, we have:

$$p_\alpha(z_a|z_b,a) \propto p_0(z_a)\exp\left[-E_\alpha(z_a,z_b,a)\right]. \tag{37}$$

We denote the real perturbation prior as: $\pi(a|z_b,c)$, according to Bayes rule, we obtain:

$$p_\alpha(a|z_a,z_b) \propto \pi(a|z_b,c)p_\alpha(z_a|z_b,a). \tag{38}$$

Combining Eq. equation 37 and Eq. equation 38, and eliminating $p_0(z_a)$, we obtain:

$$r_\alpha(a|z_a,z_b,c) = \frac{\pi(a|z_b,c)\exp(-E_\alpha(z_a,z_b,a))}{\sum_{a'\in\mathcal{A}}\pi(a'|z_b,c)\exp(-E_\alpha(z_a,z_b,a'))}. \tag{39}$$

From the perspective of mutual information, we consider conditional mutual information $I(A;Z_a|Z_b,C)$, having:

$$I(A;Z_a|Z_b,C) = H(A|Z_b,C) - H(A|Z_a,Z_b,C), \tag{40}$$

where $H(.|Z_b,C)$ and $H(.|Z_a,Z_b,C)$ denote the conditional entropy. According to the definition of entropy, we have:

$$-H(A|Z_a,Z_b,C) = \mathbb{E}\left[log\, p(a|z_a,z_b,c)\right], \tag{41}$$

where $p(a|z_a,z_b,c)$ is the real perturbation posterior distribution. Therefore we have:

$$I(A;Z_a|Z_b,C) = H(A|Z_b,C) + \mathbb{E}\left[log\, p(a|z_a,z_b,c)\right]. \tag{42}$$

We use $r_\alpha(a|z_a,z_b,c)$ to approximate $p(a|z_a,z_b,c)$, and because the property of KL-divergence that $D_{KL}(p||q) \geq 0$, we have:

$$D_{KL}(p(a|z_a,z_b,c)||r_\alpha(a|z_a,z_b,c)) \geq 0, \tag{43}$$

Therefore, we obtain:

$$I(A;Z_a|Z_b,C) \geq H(A|Z_b,C) + \mathbb{E}\left[\log r_\alpha(A|Z_a,Z_b,C)\right], \tag{44}$$

where the equality holds when $r_\alpha(a|z_a,z_b,c)$ correctly approximates the real distribution $p(a|z_a,z_b,c)$.

Thus, we have proved the formation of Theorem 2.

Next, we will talk about the relationship between the energy optimization and the conditional mutual information.

According to our optimization realized in Subsection 3.3, the energy function $E_\alpha(z_a,z_b,a)$ assigns lower energy (higher probability mass) to the compatible perturbation representations and higher energy to incompatible ones (i.e., For the definition of $r_\alpha(a|z_a,z_b,c)$ in Eq. equation 39, $E_\alpha(z_a,z_b,a)$ becomes lower while $E_\alpha(z_a,z_b,a')$ becomes higher). Therefore, the optimization of energy function leads to the increase of $\mathbb{E}\left[\log r_\alpha(A|Z_a,Z_b,C)\right]$, which means **Optimizing PerturbEnergy leading to the increase of the conditional mutual information lower bound between the perturbation and the latent representation.**

Until now, We have fully proved Theorem 2. □

A.2 Algorithm Description

A.2.1 Langevin Sampling Algorithm

We provide the Langevin sampling in Algorithm 1.

A.2.2 Optimization Algorithm

We provide the optimization process in Algorithm 2

Algorithm 1 Langevin Sampling

Require: Cellular expression x_{ctr}, basal cellular representation z_b, perturbation a, energy function $E_\alpha(z_a, z_b, a)$, initial model $q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a)$, Langevin steps K, step size η
Ensure: Negative sample z_a^-

1: Initialize $z_a^{(0)} \sim q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a)$
2: **for** $k = 0, 1, \ldots, K-1$ **do**
3: Sample Gaussian noise $\epsilon^{(k)} \sim \mathcal{N}(0, I)$
4: Update $z_a^{(k)}$ via Langevin dynamics:

$$z_a^{(k+1)} = z_a^{(k)} - \eta \nabla_{z_a} E_\alpha \left(z_a^{(k)}, z_b, a \right) + \sqrt{2\eta}\, \epsilon^{(k)}$$

5: **end for**
6: Set $z_a^- = z_a^{(K)}$
7: **return** z_a^-

Algorithm 2 Optimization Algorithm

Require: Batch cell data (x_{ctr}, x) under the same context c with perturbation a, model parameters $\{\theta, \alpha, \phi_\beta, \phi_\alpha\}$, sample size N, α learning rate ζ , and $\{\theta, \phi_\beta, \phi_\alpha\}$ learning rate ξ
Ensure: Updated model parameters $\{\theta, \alpha, \phi_\beta, \phi_\alpha\}$

1: Sample batch basal representations z_b from $q_{\phi_\beta}(z_b|x_{ctr}, c)$
2: Generate N positive batch samples $\{z_{a,i}^+\}_{i=1}^N$ via $q_{\phi_\alpha}(z_a|z_b, x, a)$
3: Generate N negative batch samples $\{z_{a,i}^-\}_{i=1}^N$ via Langevin sampling in Algorithm 1 or via $q_{\phi_\alpha}(z_a|z_b, x_{ctr}, a^-)$ with random other a^-
4: Calculating $\nabla_\alpha \mathcal{J}_{\mathrm{E}}(\alpha)$ via Eq. equation 12
5: Calculating $\{\nabla_\theta \mathcal{L}_{\mathrm{ELBO}}, \nabla_{\phi_\beta} \mathcal{L}_{\mathrm{ELBO}}, \nabla_{\phi_\alpha} \mathcal{L}_{\mathrm{ELBO}}\}$ via Eq. equation 15
6: Update α via: $\alpha \leftarrow \alpha - \zeta \nabla_\alpha \mathcal{J}_{\mathrm{E}}(\alpha)$
7: Update $\{\theta, \phi_\beta, \phi_\alpha\}$ via: $\theta/\phi_\beta/\phi_\alpha \leftarrow \theta/\phi_\beta/\phi_\alpha + \xi \nabla_{\theta/\phi_\beta/\phi_\alpha} \mathcal{L}_{\mathrm{ELBO}}$

Algorithm 3 Inference Algorithm

Require: Unperturbed cell (x_{ctr}, c), perturbation a, model parameters $\{\theta, \alpha, \phi_\beta, \phi_\alpha\}$, number of samples M
Ensure: Predicted cellular response x

1: Sample basal representation z_b from $q_{\phi_\beta}(z_b|x_{ctr}, c)$
2: Sample M perturbation representations $\{z_{a,i}\}_{i=1}^M$ from $q_{\phi_\alpha}(z_a|z_b, a)$
3: Compute energy scores $\{E_\alpha(z_{a,i}, z_b, a)\}_{i=1}^M$
4: Select $z_a^* = \arg\min_{z_{a,i}} E_\alpha(z_{a,i}, z_b, a)$
5: Generate predicted cellular response x from $p_\theta(x|z_b, z_a^*)$
6: **return** x

A.2.3 Inference Algorithm

We provide the inference process in Algorithm 3

A.3 Experimental Appendix

A.3.1 Dataset Descriptions and Settings

We evaluate on four widely used single-cell perturbation benchmarks that together span signaling, chemical and genetic interventions: Replogle Replogle et al. (2022), PBMC Yuan et al. (2026), Norman2019 Norman et al. (2019) and Tahoe100M Zhang et al. (2025). Following PerturbDiff Yuan et al. (2026), STATE Adduri et al. (2026) and VC Benchmark Mao et al. (2026), we adopt the same data sources, preprocessing pipeline and held-out splits. Summary statistics are given in Table. 4.

Table 4: *(a)* Perturbation datasets. Statistics include counts of perturbed cell across train/valid/test splits and counts of control cells, total gene vocabulary size and HVGs, and dataset diversity in terms of perturbations, cell types, and experimental batches.

Dataset	Perturbation Type	#Perturbed Cells			#Control Cells	#Genes	#HVGs	#Perturbations	#Cell Types	#Batches
		Train	Valid	Test						
Replogle	gene	572,545	4,825	26,878	39,165	6,642	2,000	2,023	4	56
PBMC	cytokine	6,657,336	150,484	2,260,453	629,701	62,710	2,000	90	18	12
Norman	gene	85,568	1,754	12,268	11,855	33,694	2,000	236	1	8
Tahoe100M	drug	89,495,239	553,076	8,790,272	2,330,156	40,352	2,000	1,137	50	14

Replogle (genetic perturbation). The Replogle dataset Replogle et al. (2022) is a landmark resource for studying single-cell genetic perturbation effects. It integrates multiple pooled CRISPR interference (CRISPRi) Gilbert et al. (2013); Nadig et al. (2025) experiments across four human cell lines: K562, RPE1, Jurkat and HepG2. It systematically repressing thousands of essential genes while measuring transcriptome-wide responses. Because the targeted genes are essential, a single knockdown often induces broad transcriptional changes that affect hundreds of downstream genes.

PBMC (cytokine signaling). The PBMC dataset Yuan et al. (2026) is a large-scale signaling perturbation atlas of peripheral blood mononuclear cells. It profiles approximately 10 million human cells collected from 12 healthy donors, covering 18 immune cell types and 90 distinct cytokine treatments alongside a PBS control condition. After quality control, about 9.7 million cells are retained, with a mean sequencing depth of roughly 31,000 reads per cell. Many cytokines elicit strong transcriptional responses, typically upregulating more than 50 genes, which makes this dataset a representative testbed for studying perturbation-induced shifts at the population level.

Tahoe100M (drug perturbation). Tahoe100M (Zhang et al., 2025) is the largest single-cell perturbation atlas released to date. It comprises over 100 million single-cell expression profiles measured across 50 human cancer cell lines exposed to more than 1,100 combinations of small-molecule drug and dosage. The atlas adopts a multiplexed cell-village design during sequencing, which substantially reduces batch effects and yields high data quality. Its scale and the breadth of its cellular contexts make it a demanding benchmark for context-dependent drug response prediction.

Norman2019 (combinatorial genetic perturbation). The Norman2019 dataset Norman et al. (2019) is a Perturb-seq screen of combinatorial CRISPR activation (CRISPRa) in K562 cells. Its value lies not in the diversity of cellular contexts but in the structure of its perturbation space: it comprises 105 single-gene activations and 131 two-gene combinations, and both constituent genes of every combination are also activated individually elsewhere in the screen. This makes it the standard benchmark for combinatorial generalization, meaning predicting the transcriptional outcome of a gene pair from the effects of its individual members. Such predictions are far from trivial, because gene pairs frequently interact non-additively: two genes whose individual effects are mild may act synergistically, and a naive additive model that simply sums their single-gene effects fails to recover the resulting phenotype. We use the filtered version of the dataset as VC Benchmark Mao et al. (2026), which contains 111,445 cells.

Preprocessing. We apply the same preprocessing pipeline as PerturbDiff Yuan et al. (2026), STATE Adduri et al. (2026) and VC Benchmark Mao et al. (2026). For PBMC and Tahoe100M, raw counts are first normalized per cell to a fixed library size(10^4), followed by a log1p transformation to rescale the gene expression values into range $[0, 1)$. For Replogle, we directly adopt the processed data released by STATE Adduri et al. (2026), which first filters cells and perturbations by on-target knockdown efficacy to improve the signal-to-noise ratio of perturbation effects, and then applies the same library-size normalization and log transformation. After normalization, we follow standard practice Wolf et al. (2018) and select the top 2,000 highly variable genes (HVGs) for each dataset from its original transcriptome vocabulary using `scanpy.pp.highly_variable_genes`. Excluding low-variance genes improves the signal-to-noise ratio without sacrificing biological fidelity, and all models are trained and evaluated in this 2,000-dimensional HVG space. Norman2019 is distributed as raw counts and no curated gene panel is available for it, so we apply the same normalization pipeline and then select the top 2,000 HVGs from its 33,694-gene vocabulary.

Train/validation/test splits. For Replogle, PBMC and Tahoe100M, we follow the hold-out protocol of PerturbDiff Yuan et al. (2026) and STATE Adduri et al. (2026), which measures generalization

to unseen cellular contexts. For each dataset, a subset of contexts is held out for testing, and 30% of the perturbations measured in those held-out contexts are moved back into the training set. This partial-coverage design ensures that every perturbation is observed somewhere during training, so that the test set isolates adaptation to a new cellular context rather than extrapolation to an unknown perturbation label. Concretely:

- **PBMC.** Four of the twelve donors are reserved for testing. 30% of the perturbations from these held-out donors are moved into training, leaving the remaining 70% for testing. Note that the model conditions on cell type (18 categories), while the hold-out is taken over donors; at test time the control cells are therefore drawn from the held-out donors themselves.
- **Tahoe100M.** PCA is first performed on the pseudobulk expression profiles of the 50 cell lines to identify distinct phenotypic clusters, from which five cell lines are selected as the held-out test set, again with 30% of their perturbations returned to training.
- **Replogle.** One cell line (HepG2) is held out for testing, and the model is trained on the remaining three cell lines together with 30% of the perturbations randomly sampled from HepG2.

Norman2019 contains a single cell line, so holding out a cellular context is degenerate. We therefore split over the perturbations themselves, following the VC benchmark Mao et al. (2026): 30% of the two-gene combinations (39 of 131) are held out as the test set, while all 105 single-gene perturbations remain in training. This is the "0 of 2 unseen" generalization setting. Each held-out combination is never observed during training, yet both of its constituent genes have been perturbed individually, which isolates the ability to compose known single-gene effects into an unseen combination, rather than confounding it with extrapolation to genes for which no perturbation data exist at all. It is also what keeps the held-out combinations representable by our model: we embed a perturbation as the sum of the embeddings of its target genes, so an unseen pair is encoded from the two single-gene rows that training has already fitted. A per-label lookup table, by contrast, would have no entry for such a pair and could not make a prediction at all.

Control cells remain in the training pool under all three protocols: they are the unperturbed reference population that the model is given at inference time, and they are therefore counted separately from the train/validation/test splits. Finally, we hold out 2% of the training cells uniformly at random as a validation set. This set is used only for checkpoint selection; all reported results come from the held-out contexts in the test split.

A.3.2 Baseline Settings

Baseline models

Linear methods. The linear baseline Ahlmann-Eltze et al. (2024) predicts a perturbed cell as the control cell plus the mean expression shift of that perturbation measured on the training split, $\hat{x} = x_{ctrl} + \Delta_a$, with the shift Δ_a taken against the controls of the contexts in which the perturbation was observed. Despite its simplicity, this predictor is a strong baseline on average-expression metrics and serves as the linear reference point of the comparison. **scGPT** (Cui et al., 2024) replaces the raw expression profile with a pretrained cell representation: we use the released `scGPT_human` checkpoint (12 layers, d_{model}=512, 8 heads, 51.9M parameters, pretrained on 33M cells), keep it frozen in evaluation mode, and train only a lightweight head on top, $\hat{x} = x_{\text{ctrl}} + \text{MLP}([u - z])$, where z is the `<cls>` embedding of the control cell and u the perturbation embedding. Cells are tokenised exactly as during scGPT's pretraining: gene symbols mapped through its 60,697-entry vocabulary, nonzero genes only, per-cell quantile binning into 51 bins, and a prepended `<cls>` token. The head's output layer is zero-initialised, so training starts from "predict no change". This row answers a narrower question than the others: whether a representation pretrained on external data already encodes how a cell responds to a perturbation.

VAE-based methods. All the VAE-based methods are implemented via their official codes. **scGen** Lotfollahi et al. (2019) trains an unconditional VAE on all cells. After training, the perturbation vector $\Delta_a = \text{mean}\{z_a\} - \text{mean}\{z_{ctr}\}$ is measured on the training split, where z_a is the representation of perturbed cells and z_{ctr} is the representations of controlled cells. And prediction is $\hat{x} = \text{Dec}(\text{Enc}(x_{\text{ctrl}}) + \Delta_a)$. **CPA** (Lotfollahi et al., 2023) decomposes a cell into a basal state

plus additive perturbation and covariate embeddings, $z = \mathrm{Enc}(x) + u_a + u_c$, and enforces the decomposition with adversarial discriminators trained to disentangle the perturbation and the covariates of the basal latent. Prediction swaps the perturbation embedding to the corresponding one. **stVAE** (Russkikh et al., 2020) is a style-transfer VAE: expression is decoupled into a perturbation-independent content latent and a perturbation-specific style vector, and prediction decodes the sum of the control cell's content and the target perturbation's style. **PerturbedVAE** (Jiang et al., 2026) splits the latent into a perturbation-invariant part z_i and a perturbation-responsive part z_v, and makes the split persist with a contrastive term that anchors z_i to the corresponding control cell. The prediction mechanism is a learned conditional prior $p(z_v \mid z_i, u)$, which supplies the response at inference time. As their official implementation suggests, CPA, stVAE, and PerturbedVAE are trained on observed cells and scGen is trained unconditionally. For a fair comparison, we adopt their official VAE architecture but unify the latent embedding size the same as ours: 512.

Diffusion-based methods. PerturbDiff (Yuan et al., 2026) follows its official implementation, it runs a denoising diffusion process over the cell population rather than over individual cells. The denoiser is an MM-DiT stack (4 blocks, 8 heads, latent dimension is setted as 512) with joint attention over the control and perturbed token streams, a single shared timestep noises the whole population so that the diffused object remains a distribution, and the denoising objective is the MMD between the denoised and the real population. Inference uses 50 DDIM steps as it general leads to excellent performance. **Squidiff** (He et al., 2026) runs a conditional DDIM in a VAE semantic latent (dimension of z is adjusted to 512 as other models) with an MLP denoiser and sinusoidal timestep embeddings, and decodes the sampled latent. Its autoencoder is warmed-up trained alone for the first 4,000 steps, after which the encoder is frozen so that the latent diffusion trains against a fixed target.

Flow-based methods. CellFlow Klein et al. (2025) learns a conditional velocity field that transports the control population to the perturbed one. Training pairs the two minibatches by exact minibatch optimal transport under a squared-Euclidean cost, interpolates $x_t = (1-t)x_0 + tx_1$ with $t \sim \mathcal{U}(0,1)$, and regresses $v_\theta(x_t, t, \mathrm{cond})$ onto $x_1 - x_0$, where the conditioning combines the perturbation and context embeddings with the pooled control population. Inference integrates the field from $t{=}0$ to $t{=}1$ with a Heun solver in 50 steps.

Transformer-based method. STATE Adduri et al. (2026) is a bidirectional transformer over sets of cells, where the perturbation embedding is added to every token and the model predicts the transition from each control cell. We reproduce the State Embedding (SE), State Transition (ST), and adapt the embedding size to 512, where SE uses ESM2 embedding and project into 512 dimension. For Replogle, PBMC and Tahoe100M, we use the the same training dataset for SE: CellxGene Program et al. (2025) with the same reconstruction target. The learning rate is set as 1×10^{-5}, weight decay is set as 0.01, batch size is set as 128 and epoch number is 12. As for Norman2019, SE is pretrained on its training set with the same training settings but 5 epochs.

Training settings

All the baselines follow the same data preprocess and split(train/validation/test) settings as described in Appendix A.3.1. We use the same condition sampler and random seed for all the methods: A batch is n blocks of `set_size` cells, where each block pairs control cells of a context with real perturbed cells of one (context, perturbation) condition. Every model therefore sees the same 512 cells per step (256 on Norman2019). As for the maximum training step budget, we set 20,000 steps on Norman2019, 40,000 on Replogle, 60,000 on PBMC and 160,000 on Tahoe100M for all the methods. All models are optimised with Adam (weight decay: 10^{-6}) under their own best learning rate. The loss function and corresponding settings of each baseline follows its official suggestion. A cosine learning-rate decay is used for PerturbDiff, Squidiff, CellFlow and STATE, and a constant rate otherwise. For the two diffusion baselines and CellFlow we additionally keep an exponential moving average (EMA) of the weights (decay 0.999) and validate, checkpoint and evaluate that average, since EMA is part of how diffusion and flow-matching models are trained in practice and their reported sample quality is not what the raw iterate gives. Model-specific settings are listed in Table. 5.

Validation and checkpoint selection

Table 5: **Model-specific training settings.** All other settings including batch shape, step budget, optimiser family, gradient clipping, checkpoint selection and evaluation are shared across models and identical to those used for PerturbEnergy.

Model	Family	LR	EMA	Model-specific settings
Linear	linear	—	—	closed form; no gradient training
scGPT	foundation	1×10^{-3}	—	encoder frozen; head $512{+}512\to1024\to G$
scGen	VAE	1×10^{-3}	—	$z=512$, hidden 800×2, no conditioning
CPA	VAE	1×10^{-3}	—	$z=512$, adv. weight 50, 3 adversary steps
stVAE	VAE	1×10^{-3}	—	$z=512$, latent loss 1.0, VAE loss 0.1
PerturbedVAE	VAE	1×10^{-3}	—	$z_i=z_v=512$, contrastive weight 1.0
PerturbDiff	diffusion	3×10^{-4}	0.999	$z=512$, 4 MM-DiT blocks, 8 heads, $T=1000$
Squidiff	diffusion	1×10^{-3}	0.999	$z=512$, $T=1000$, 4k-step AE warm-up
CellFlow	flow matching	1×10^{-3}	0.999	Solver 50 steps, depth 4, OT coupling
STATE	transformer	3×10^{-4}	—	$z=512$, depth 4, 8 heads, set size 128

Validation is identical for every baseline model and PerturbEnergy. Every 2,500 steps we validate on the validation batches where the score is the Mean Squared Error (MSE) of the predicted perturbation mean against the real one. The checkpoint with the lowest validation score is the one evaluated on the test split; for models with EMA, the averaged weights are the ones both validated and saved.

A.3.3 METRICS

Metric Definitions

Preliminary Notation for Metrics Description. A *condition* $\lambda=(c,a)$ pairs a cellular context c with a perturbation a; the test split contains M such conditions, and each metric below is computed per condition and then averaged over the M of them. For condition λ, we note $X^\lambda\in\mathbb{R}^{N\times G}$ for the real perturbed cells, $\widetilde{X}^\lambda\in\mathbb{R}^{n\times G}$ for the control cells of the same context c, and $\hat{X}^\lambda\in\mathbb{R}^{n\times G}$ for the model's perturbation prediction, over $G=2{,}000$ highly variable genes indexed by $g\in\{1,\dots,G\}$. A subscript i on a matrix denotes its i-th row, i.e. a single cell in $\mathbb{R}^G$. Inference maps control cells to perturbed ones, so the prediction contains the same number of cells as the control population, n. Pseudobulk profiles are the column means, which are vectors in $\mathbb{R}^G$:

$$\mu=\tfrac{1}{N}\textstyle\sum_i X_i^\lambda,\qquad \widetilde{\mu}=\tfrac{1}{n}\textstyle\sum_i \widetilde{X}_i^\lambda,\qquad \hat{\mu}=\tfrac{1}{n}\textstyle\sum_i \hat{X}_i^\lambda, \tag{45}$$

the ground true perturbation effect is:

$$\Delta=\mu-\widetilde{\mu}, \tag{46}$$

the model predicted perturbation effect is:

$$\hat{\Delta}=\hat{\mu}-\widetilde{\mu}. \tag{47}$$

Since the data is log1p-normalised, Δ and $\hat{\Delta}$ are log fold changes.

Differentially expressed gene (DEG). The DEG set is derived from the real data so all methods are scored on the same genes. Each gene is tested with a two-sided Wilcoxon rank-sum (Mann–Whitney U) test between X^λ and $\widetilde{X}^\lambda$; genes are ordered by ascending p-value with $|\Delta_g|$, and $\mathcal{S}_k$ denotes the top-k genes.

HVG Group Metrics

Correlation-based metrics. These metrics ask how closely the predicted expression tracks the ground truth (correlation), which has two forms: measured on the expression level ($\hat{\mu}$ against μ) corresponding to the overall transcriptional state, and measured on the perturbation effect ($\hat{\Delta}$ against Δ). The metrics we use are defined as follows:

PCC: Pearson correlation between the predicted and real pseudobulk profiles; measures agreement in overall expression level. Let $\mathrm{corr}(\cdot,\cdot)$ denote the Pearson correlation:

$$\mathrm{PCC} = \mathrm{corr}(\hat{\mu}, \mu) = \frac{\sum_g (\hat{\mu}_g - \bar{\hat{\mu}})(\mu_g - \bar{\mu})}{\sqrt{\sum_g (\hat{\mu}_g - \bar{\hat{\mu}})^2}\sqrt{\sum_g (\mu_g - \bar{\mu})^2}} \tag{48}$$

PDCorr: the same Pearson correlation taken on the perturbation effect rather than the level, so an accurate baseline expression cannot inflate it. Here $\bar{\Delta}$ and $\bar{\hat{\Delta}}$ are the scalar means of the two effect vectors.

$$\mathrm{PDCorr} = \mathrm{corr}\Big(\hat{\Delta}, \Delta\Big) = \frac{\sum_g (\hat{\Delta}_g - \bar{\hat{\Delta}})(\Delta_g - \bar{\Delta})}{\sqrt{\sum_g (\hat{\Delta}_g - \bar{\hat{\Delta}})^2}\sqrt{\sum_g (\Delta_g - \bar{\Delta})^2}} \tag{49}$$

$\mathbf{R^2}$: fraction of variance in the real profile explained by the prediction, relative to the empirical mean baseline.

$$R^2 = 1 - \frac{\sum_g (\mu_g - \hat{\mu}_g)^2}{\sum_g (\mu_g - \bar{\mu})^2} \tag{50}$$

LFCSpear: Spearman rank correlation ρ_s of the log fold changes, that is the Pearson correlation computed on the ranks of the entries instead of their values, where $\mathrm{rank}(v) \in \mathbb{R}^G$ assigns each gene its position when v is sorted. Unlike PDCorr it depends only on the ordering of genes by effect, not on the magnitudes.

$$\mathrm{LFCSpear} = \rho_s\Big(\hat{\Delta}, \Delta\Big) = \mathrm{corr}\Big(\mathrm{rank}(\hat{\Delta}), \mathrm{rank}(\Delta)\Big) \tag{51}$$

LogFC_cos: cosine similarity of the two effect vectors, i.e. whether the response points in the right direction irrespective of its scale.

$$\mathrm{LogFC_cos} = \frac{\langle \hat{\Delta}, \Delta \rangle}{\|\hat{\Delta}\|_2 \, \|\Delta\|_2} = \frac{\sum_g \hat{\Delta}_g \Delta_g}{\sqrt{\sum_g \hat{\Delta}_g^2}\sqrt{\sum_g \Delta_g^2}} \tag{52}$$

DirAgr: fraction of genes whose direction of change is predicted correctly. The average is taken over $\mathcal{G}^+ = \{g : \Delta_g \neq 0\}$: with log-normalised counts many genes have a true effect of exactly zero in a small population, and those can only be "agreed" with by predicting exactly zero, which would pull every model towards the do-nothing baseline. Let $\mathrm{sign}(\cdot) \in \{-1, 0, +1\}$ denotes the sign of a scalar and $\mathbb{1}[\cdot]$ denotes the indicator function equal to one when its argument holds and zero otherwise.

$$\mathrm{DirAgr} = \frac{1}{|\mathcal{G}^+|} \sum_{g \in \mathcal{G}^+} \mathbb{1}\Big[\mathrm{sign}(\hat{\Delta}_g) = \mathrm{sign}(\Delta_g)\Big] \tag{53}$$

PDS-ℓ_1/ℓ_2/cos: perturbation discrimination score. A prediction should resemble its own ground truth more than any other condition's. Let $\hat{\Delta}_\lambda$ be the predicted effect for condition λ and Δ_p the real effect of condition p. The rank $r_\lambda \in \{0, \ldots, M-1\}$ counts how many of the other $M-1$ ground truths lie closer to $\hat{\Delta}_\lambda$ than its own does, and dividing by $M-1$ normalises it to $[0, 1]$:

$$\mathrm{PDS} = \frac{1}{M} \sum_{\lambda=1}^{M} \frac{r_\lambda}{M-1}, \qquad r_\lambda = \sum_{p \neq \lambda} \mathbb{1}\Big[d(\hat{\Delta}_\lambda, \Delta_p) < d(\hat{\Delta}_\lambda, \Delta_\lambda)\Big], \tag{54}$$

with distance the ℓ_1, ℓ_2 or cosine distance. **Lower is better**: 0 means every prediction is nearest its own ground truth and 0.5 is chance. (Works reporting $1 - \mathrm{PDS}$ invert the direction.)

DEG distinction metrics. These metrics ask a different question from the correlation measures: whether the model can identify the correct DEGs. Let $\mathcal{S}_k$ be the true top-k DE genes and $\hat{\mathcal{S}}_k$ the k genes with the largest predicted $|\hat{\Delta}_g|$; we use $k = 50$. These metrics include: DEPrec, DERec, DEOverlap, AUROC and AUPRC.

DEPrec, DERec, DEOverlap: precision, recall and Jaccard index of the retrieved DEG set, where $|\cdot|$ is set cardinality and $\cap, \cup$ are intersection and union. Because the two sets are built with the same size ($|\hat{\mathcal{S}}_k| = |\mathcal{S}_k| = k$), precision and recall coincide here.

$$\mathrm{DEPrec} = \frac{|\mathcal{S}_k \cap \hat{\mathcal{S}}_k|}{|\hat{\mathcal{S}}_k|}, \quad \mathrm{DERec} = \frac{|\mathcal{S}_k \cap \hat{\mathcal{S}}_k|}{|\mathcal{S}_k|}, \quad \mathrm{DEOverlap} = \frac{|\mathcal{S}_k \cap \hat{\mathcal{S}}_k|}{|\mathcal{S}_k \cup \hat{\mathcal{S}}_k|} \tag{55}$$

AUROC, AUPRC: Genes in $\mathcal{S}_k$ are labelled positive, all remaining $G-k$ genes negative, and every gene is scored by $|\hat{\Delta}_g|$; we report the area under the ROC curve and the average precision.

Distance-based metrics. These metrics evaluate the distance between the prediction and real expression. We use Root Mean Square Error (RMSE), Mean Square Error (MSE), Mean Absolute Error (MAE), Energy Distance (EDist) and Sinkhorn Divergence (Sinkhorn).

RMSE, MSE, MAE: error between the predicted and real pseudobulk.

$$\mathrm{MSE} = \frac{1}{G}\sum_g (\hat{\mu}_g - \mu_g)^2, \quad \mathrm{RMSE} = \sqrt{\mathrm{MSE}}, \quad \mathrm{MAE} = \frac{1}{G}\sum_g |\hat{\mu}_g - \mu_g| \tag{56}$$

EDist: energy distance between the two populations, zero if and only if the distributions coincide.

$$\mathrm{EDist} = \frac{2}{nN}\sum_{i,j} \|\hat{x}_i - x_j\|_2 - \frac{1}{n^2}\sum_{i,j} \|\hat{x}_i - \hat{x}_j\|_2 - \frac{1}{N^2}\sum_{i,j} \|x_i - x_j\|_2 \tag{57}$$

Sinkhorn: entropic optimal-transport divergence, debiased so that $\mathrm{S}(X, X) = 0$:

$$\mathrm{S}_\varepsilon(\hat{X}, X) = \mathrm{OT}_\varepsilon(\hat{X}, X) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(\hat{X}, \hat{X}) - \tfrac{1}{2}\mathrm{OT}_\varepsilon(X, X), \tag{58}$$

$$\mathrm{OT}_\varepsilon(A, B) = \min_{P \in \Pi(a,b)} \langle P, C\rangle + \varepsilon\, \mathrm{KL}(P \,\|\, ab^\top), \qquad C_{ij} = \|A_i - B_j\|_2^2/s, \tag{59}$$

with uniform marginals, $\varepsilon = 0.05$, and s the mean squared pairwise distance within the control population, which makes the value comparable across datasets. Both distances are evaluated on a fixed random subsample of the populations (512 cells for EDist, 256 for Sinkhorn) to bound their quadratic cost.

DEG Group Metrics

This metric group evaluates how well a model predict the biologically meaningful gene expression specifically influenced by the perturbation. Different from the HVG group describing the general expression pattern, the DEG group is directly related to the perturbation. Since perturbation leaves most of the transcriptome nearly untouched, so genome-wide averages are dominated by genes that carry no signal and a model can score well while missing the response entirely. The DEG group therefore restricts the error and correlation measures to the true top-k DE genes, $k \in \{50, 100, 200\}$. With $\mathcal{S}_k$ fixed in advance from the real data, this group no longer asks whether the responding genes were found. We calculate the RMSE, MAE, PCC and R^2 on the DEGs.

$$\mathrm{MSE}^{(k)} = \frac{1}{k}\sum_{g \in \mathcal{S}_k} (\hat{\mu}_g - \mu_g)^2, \quad \mathrm{RMSE}^{(k)} = \sqrt{\mathrm{MSE}^{(k)}}, \qquad \mathrm{MAE}^{(k)} = \frac{1}{k}\sum_{g \in \mathcal{S}_k} |\hat{\mu}_g - \mu_g|, \tag{60}$$

$$\mathrm{PCC}^{(k)} = \mathrm{corr}(\hat{\mu}_{\mathcal{S}_k}, \mu_{\mathcal{S}_k}), \qquad R^{2(k)} = 1 - \frac{\sum_{g \in \mathcal{S}_k} (\mu_g - \hat{\mu}_g)^2}{\sum_{g \in \mathcal{S}_k} (\mu_g - \bar{\mu}_{\mathcal{S}_k})^2}. \tag{61}$$

A.3.4 Other VAE Backbone Integration

Since our PerturbEnergy is backbone-agnostic, we integrate it with two additional VAE backbones, scGen and PerturbedVAE, with the results reported in Table 6. Since scGen does not explicitly learn a separate perturbation representation, we adapt the energy function from $E_\alpha(z_a, z_b, a)$ to $E_\alpha(z, a)$, which computes the cosine similarity between the latent representation and the perturbation embedding. For PerturbedVAE, we directly apply $E_\alpha(z_a, z_b, a)$ to its basal and perturbation representations. Both integrations preserve the original architectures and training objectives, with

only the energy-based optimization term in Eq. equation 12 added. The results show that integrating PerturbEnergy consistently improves prediction performance, particularly for scGen. Specifically, PerturbEnergy reduces the top-50 DEG RMSE of scGen by 18.6% on Replogle (from 0.2336 to 0.1902) and 24.1% on Tahoe100M (from 0.4462 to 0.3387). These substantial improvements suggest that explicitly incorporating perturbation-specific information is particularly beneficial for models without a dedicated perturbation representation. PerturbEnergy also improves PerturbedVAE, although the gains are relatively smaller, potentially because PerturbedVAE already captures some perturbation-specific information. Collectively, these results demonstrate the flexibility of PerturbEnergy across different VAE backbones and its effectiveness in improving cellular response prediction.

Table 6: Backbone transfer: RMSE (↓) over the true top-K DEGs when the PerturbEnergy objective is applied to two other backbones. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Within each backbone block, the best entry per column is in **bold**.

Method	Replogle			Tahoe100M		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
scGen	0.2336 ±0.0078	0.1992 ±0.0065	0.1661 ±0.0051	0.4462 ±0.0081	0.3818 ±0.0059	0.3018 ±0.0046
+ PerturbEnergy (Lang)	0.1931 ±0.0064	0.1631 ±0.0052	0.1356 ±0.0041	0.3457 ±0.0061	0.3023 ±0.0044	0.2421 ±0.0034
+ PerturbEnergy (Mis)	**0.1902** ±0.0063	**0.1607** ±0.0051	**0.1341** ±0.0040	**0.3387** ±0.0059	**0.2966** ±0.0043	**0.2379** ±0.0033
PerturbedVAE	0.2130 ±0.0071	0.1817 ±0.0058	0.1514 ±0.0047	0.3984 ±0.0075	0.3476 ±0.0055	0.2791 ±0.0042
+ PerturbEnergy (Lang)	**0.1966** ±0.0058	**0.1692** ±0.0048	0.1424 ±0.0038	0.3892 ±0.0073	**0.3324** ±0.0053	**0.2653** ±0.0040
+ PerturbEnergy (Mis)	0.1969 ±0.0059	0.1694 ±0.0049	**0.1422** ±0.0039	**0.3759** ±0.0068	0.3327 ±0.0050	0.2692 ±0.0039

Method	PBMC			Norman2019		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
scGen	0.0960 ±0.0049	0.0787 ±0.0040	0.0625 ±0.0031	0.1729 ±0.0102	0.1404 ±0.0082	0.1108 ±0.0065
+ PerturbEnergy (Lang)	0.0706 ±0.0021	0.0600 ±0.0017	0.0495 ±0.0014	**0.1560** ±0.0127	**0.1265** ±0.0099	**0.0990** ±0.0073
+ PerturbEnergy (Mis)	**0.0703** ±0.0022	**0.0597** ±0.0018	**0.0493** ±0.0015	0.1645 ±0.0137	0.1329 ±0.0107	0.1030 ±0.0079
PerturbedVAE	0.1071 ±0.0032	0.0925 ±0.0025	0.0777 ±0.0020	0.1400 ±0.0123	0.1186 ±0.0100	0.0953 ±0.0075
+ PerturbEnergy (Lang)	**0.0987** ±0.0031	**0.0849** ±0.0024	**0.0710** ±0.0019	**0.1312** ±0.0115	0.1109 ±0.0093	0.0919 ±0.0071
+ PerturbEnergy (Mis)	0.1007 ±0.0028	0.0880 ±0.0022	0.0740 ±0.0017	0.1379 ±0.0108	**0.1072** ±0.0087	**0.0897** ±0.0066

A.3.5 Influence of Training Scale and Model Dimension

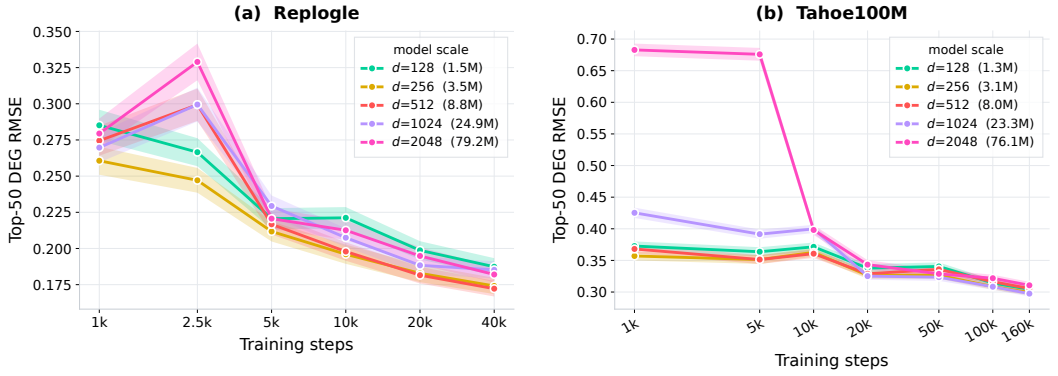

Figure 5: Influence of training steps and model dimension on Replogle and Tahoe100M.

We investigate the effects of training scale and latent dimension on Replogle and Tahoe100M, using RMSE on the top-50 DEGs as the evaluation metric. The results are shown in Figure 5. We observe that insufficient training steps consistently degrade performance across all latent dimensions and datasets, suggesting that learning generalizable perturbation effects requires adequate training. Moreover, increasing the latent dimension does not necessarily improve performance, with the best results generally achieved when d ranges from 256 to 1024. For example, $d = 512$ generally achieves the best performance on Replogle. One possible explanation is that larger latent dimensions increase model capacity but do not necessarily improve the ability to capture perturbation-specific

effects. These findings also suggest that relying on model expressiveness to capture perturbation effect is unstable.

A.3.6 VISUALIZATION

We visualize the latent representations learned by PerturbEnergy and two VAE-based methods, scGen and PerturbedVAE, alongside the original gene expression profiles. We use the test set of Tahoe100M to visualize and the UMAP visualization results are shown in Figure 6. We observe that perturbation-specific information is largely obscured in the original expression space. Although scGen captures some perturbation effects, its representations remain largely indistinguishable across perturbations. PerturbedVAE improves the separation of perturbation-specific representations, but some perturbations, such as Betamethasone dipropionate and Triamcinolone, remain mixed. In contrast, PerturbEnergy exhibits clearer separation between perturbations, demonstrating its ability to preserve perturbation-specific information in the latent space.

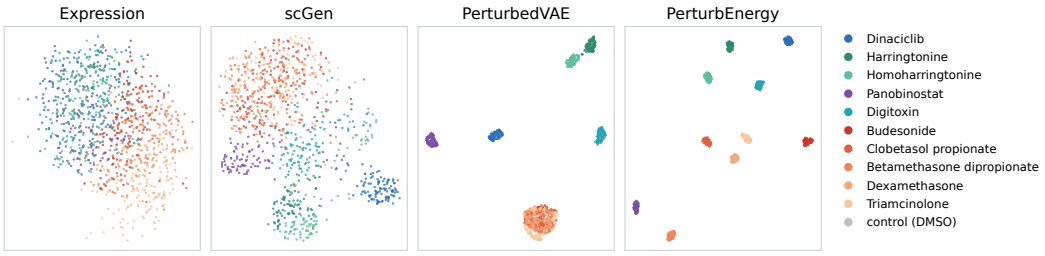

Figure 6: Visualization of representation in Tahoe100M.

A.3.7 OTHER METRICS ON DEG GROUP

We report the MAE, PCC and R^2 on Top-50/100/200 differential expression genes (DEGs) in Table 7, Table 8 and Table 9 respectively. Firstly, about our method, we have the consistent observation with the RMSE where PerturbEnergy achieves the top-2 performance on every dataset except Norman2019 even under simple MLP implementations. Specifically, in terms of MAE, the margins over the strongest baseline are largest. For example, PerturbEnergy-Mis reduces MAE from 0.1675 (Squidiff) to 0.1434 on Replogle (16.8% improvement); from 0.2648 (stVAE) to 0.2495 on Tahoe100M; on PBMC, PerturbEnergy-Lang reduces it from 0.0511 (stVAE) to 0.0453 (11.4% improvement). As for PCC and R^2, we have the same ordering observations. Secondly, for the remaining baselines, we find that: (i) Linear methods still struggle on the DEG-related prediction, for example, in MAE it achives 0.2114 compared to the best 0.1434. This again shows the regression is difficult to capture the perturbation effects; (ii) For STATE, PerturbDiff, CellFlow and Squidiff, their performance also vary a lot. In terms of MAE, PerturbDiff achieves best on Norman2019 but struggles on Replogle; Squidiff achieves competitive performance on Replogle but struggles on Tahoe100M; Cellflow perform well on PBMC and Norman2019 but get declined performance on Tahoe100M. These observations show that relying on model expressiveness to capture perturbation effect is unstable. (iii) For the representation learning baselines, especially these learn the perturbation-aware representations, we find they generally can achieve competitive performance. For example in top-50 MAE, PerturbedVAE gets 0.1812 on Replogle and 0.0703 on PBMC, which is the best among other baselines. This prove the effectiveness of modeling perturbation information.

A.3.8 RESULTS ON HVG GROUP

Results on Replogle Dataset

The HVG group results on Replogle is shown in Table 10, Table 11 and Table 12. Table 10 summarizes the correlation based measures, we find that our PerturbEnergy generally achive the best performance, which shows its effectiveness of capturing the whole transcriptomic pattern of cellular response. Table 11 is the results of differential expression gene (DEG) measures. DEG recognition is directly related to perturbation. We find that our PerturbEnergy consistently outperform

Table 7: MAE (↓) over the true top-K DEGs. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	Replogle			Tahoe100M		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
Linear	0.2282 ±0.0076	0.1963 ±0.0062	0.1634 ±0.0049	0.3788 ±0.0069	0.3169 ±0.0046	0.2246 ±0.0034
Linear (scGPT)	0.2114 ±0.0076	0.1757 ±0.0060	0.1418 ±0.0046	0.3092 ±0.0057	0.2624 ±0.0039	0.1884 ±0.0029
CellFlow	0.1925 ±0.0063	0.1615 ±0.0051	0.1306 ±0.0039	0.3280 ±0.0059	0.2767 ±0.0039	0.1975 ±0.0029
Squidiff	0.1675 ±0.0047	0.1433 ±0.0037	0.1205 ±0.0029	0.3077 ±0.0061	0.2785 ±0.0042	0.2116 ±0.0030
PerturbDiff	0.2304 ±0.0092	0.1904 ±0.0075	0.1519 ±0.0059	0.2878 ±0.0062	0.2415 ±0.0042	0.1748 ±0.0031
STATE	0.1704 ±0.0058	0.1424 ±0.0046	0.1162 ±0.0035	0.2787 ±0.0037	0.2394 ±0.0025	0.1734 ±0.0019
scGen	0.1985 ±0.0071	0.1665 ±0.0058	0.1347 ±0.0044	0.3654 ±0.0076	0.3040 ±0.0051	0.2165 ±0.0038
CPA	0.3164 ±0.0126	0.2615 ±0.0105	0.2067 ±0.0081	0.4564 ±0.0072	0.4047 ±0.0047	0.2977 ±0.0037
stVAE	0.1797 ±0.0065	0.1455 ±0.0049	0.1159 ±0.0036	0.2648 ±0.0051	0.2263 ±0.0033	0.1646 ±0.0025
PerturbedVAE	0.1812 ±0.0066	0.1509 ±0.0054	0.1216 ±0.0042	0.3160 ±0.0068	0.2666 ±0.0047	0.1918 ±0.0034
PerturbEnergy-Lang	0.1541 ±0.0055	0.1281 ±0.0044	0.1044 ±0.0033	0.2503 ±0.0045	0.2131 ±0.0030	0.1550 ±0.0022
PerturbEnergy-Mis	**0.1434** ±0.0051	**0.1196** ±0.0040	**0.0979** ±0.0030	**0.2495** ±0.0045	**0.2115** ±0.0030	**0.1531** ±0.0022

Method	PBMC			Norman2019		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
Linear	0.0836 ±0.0042	0.0672 ±0.0033	0.0515 ±0.0025	0.3285 ±0.0244	0.2335 ±0.0189	0.1506 ±0.0127
Linear (scGPT)	0.0780 ±0.0041	0.0619 ±0.0032	0.0466 ±0.0024	0.2325 ±0.0072	0.1461 ±0.0057	0.1086 ±0.0039
CellFlow	0.0654 ±0.0021	0.0511 ±0.0015	0.0376 ±0.0011	0.1264 ±0.0127	0.0977 ±0.0102	0.0699 ±0.0071
Squidiff	0.0855 ±0.0044	0.0692 ±0.0034	0.0532 ±0.0026	0.1513 ±0.0076	0.1257 ±0.0062	0.0966 ±0.0048
PerturbDiff	0.0568 ±0.0025	0.0455 ±0.0018	0.0349 ±0.0014	**0.0950** ±0.0068	**0.0754** ±0.0051	**0.0563** ±0.0036
STATE	0.0738 ±0.0028	0.0580 ±0.0022	0.0431 ±0.0017	0.1138 ±0.0100	0.0885 ±0.0074	0.0644 ±0.0048
scGen	0.0712 ±0.0045	0.0546 ±0.0036	0.0396 ±0.0027	0.1312 ±0.0080	0.1019 ±0.0063	0.0738 ±0.0046
CPA	0.1009 ±0.0057	0.0769 ±0.0043	0.0559 ±0.0031	0.3629 ±0.0250	0.2580 ±0.0191	0.1654 ±0.0128
stVAE	0.0511 ±0.0018	0.0403 ±0.0014	0.0306 ±0.0010	0.1664 ±0.0134	0.1228 ±0.0101	0.0842 ±0.0066
PerturbedVAE	0.0703 ±0.0026	0.0559 ±0.0019	0.0429 ±0.0014	0.1126 ±0.0108	0.0923 ±0.0085	0.0690 ±0.0058
PerturbEnergy-Lang	**0.0453** ±0.0016	**0.0362** ±0.0012	**0.0282** ±0.0009	0.1129 ±0.0095	0.0876 ±0.0068	0.0634 ±0.0045
PerturbEnergy-Mis	0.0467 ±0.0016	0.0376 ±0.0012	0.0292 ±0.0009	0.1075 ±0.0091	0.0842 ±0.0065	0.0614 ±0.0044

other baselines with a clear margin, which validates the effectiveness of modeling the perturbation-specific information. Table 12 summarize the distance-based measures. Our PerturbEnergy is also competitive with other baselines.

Results on PBMC Dataset

The HVG group results on PBMC is shown in Table 13, Table 14 and Table 15. From the correlation and distance related measures in Table 13 and Table 15 respectively, we have the same observations with Replogle, where PerturbEnergy outperforms or remains competitive to other baselines, showing its ability to capture the general expression patterns. From the DE-recoginition results in Table 14, we also find that PerturbEnergy outperform others especially VAE baselines (0.2473 vs 0.2208). This benefit comes from modeling perturbation-specific information.

Results on Norman2019 Dataset

The HVG group results on Norman2019 is shown in Table 16, Table 17 and Table 18. On this dataset we find a relative different pattern from Replogle and PBMC where diffusion-based methods like PerturbDiff and CellFlow ahieve the best performance on correlation or distance based measures (Table 16 and Table 18). Norman2019 generally contains duel-genetic perturbations and this diffusion process is good at capturing the complex pattern co-influenced by duel-genetic perturbations. Moreover, increasing perturbation information remains effective as we find that in DE related measures PerturbEnergy remains the best compared to other baselines (Table 17).

Results on Tahoe100M Dataset

The HVG group results on Tahoe100M is shown in Table 19, Table 20 and Table 21. The results show general the same pattern as in Replogle and PBMC.

Table 8: PCC (↑) over the true top-K DEGs. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	Replogle			Tahoe100M		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
Linear	0.9611 ±0.0024	0.9676 ±0.0018	0.9712 ±0.0015	0.9202 ±0.0031	0.9323 ±0.0022	0.9456 ±0.0015
Linear (scGPT)	0.9692 ±0.0020	0.9753 ±0.0015	0.9789 ±0.0012	0.9406 ±0.0026	0.9480 ±0.0019	0.9573 ±0.0013
CellFlow	0.9759 ±0.0017	0.9809 ±0.0012	0.9840 ±0.0009	0.9372 ±0.0030	0.9460 ±0.0021	0.9563 ±0.0015
Squidiff	0.9760 ±0.0013	0.9795 ±0.0009	0.9813 ±0.0008	0.9116 ±0.0034	0.9145 ±0.0025	0.9250 ±0.0019
PerturbDiff	0.9644 ±0.0025	0.9720 ±0.0019	0.9768 ±0.0015	0.9443 ±0.0036	0.9521 ±0.0026	0.9610 ±0.0019
STATE	0.9780 ±0.0015	0.9824 ±0.0011	0.9853 ±0.0009	0.9516 ±0.0017	0.9569 ±0.0013	0.9642 ±0.0009
scGen	0.9713 ±0.0019	0.9772 ±0.0014	0.9808 ±0.0011	0.9219 ±0.0035	0.9333 ±0.0025	0.9455 ±0.0018
CPA	0.9405 ±0.0040	0.9534 ±0.0030	0.9616 ±0.0025	0.7536 ±0.0066	0.7615 ±0.0048	0.7917 ±0.0037
stVAE	0.9751 ±0.0018	0.9809 ±0.0013	0.9843 ±0.0010	0.9507 ±0.0025	0.9566 ±0.0018	0.9640 ±0.0013
PerturbedVAE	0.9766 ±0.0017	0.9815 ±0.0012	0.9843 ±0.0010	0.9334 ±0.0032	0.9409 ±0.0024	0.9506 ±0.0017
PerturbEnergy-Lang	0.9805 ±0.0016	0.9847 ±0.0011	0.9872 ±0.0008	0.9566 ±0.0019	0.9618 ±0.0014	0.9683 ±0.0010
PerturbEnergy-Mis	**0.9824** ±0.0013	**0.9860** ±0.0009	**0.9881** ±0.0007	**0.9567** ±0.0018	**0.9619** ±0.0013	**0.9684** ±0.0010

Method	PBMC			Norman2019		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
Linear	0.9394 ±0.0039	0.9490 ±0.0029	0.9562 ±0.0021	0.9427 ±0.0075	0.9586 ±0.0052	0.9692 ±0.0046
Linear (scGPT)	0.9534 ±0.0032	0.9619 ±0.0024	0.9684 ±0.0018	0.9548 ±0.0007	0.9655 ±0.0006	0.9711 ±0.0005
CellFlow	0.9685 ±0.0018	0.9742 ±0.0011	0.9787 ±0.0008	0.9906 ±0.0020	0.9927 ±0.0014	0.9938 ±0.0014
Squidiff	0.9289 ±0.0044	0.9403 ±0.0031	0.9479 ±0.0023	0.9850 ±0.0015	0.9869 ±0.0012	0.9885 ±0.0010
PerturbDiff	0.9668 ±0.0023	0.9715 ±0.0016	0.9752 ±0.0011	**0.9949** ±0.0007	**0.9957** ±0.0006	**0.9964** ±0.0005
STATE	0.9594 ±0.0023	0.9673 ±0.0015	0.9730 ±0.0011	0.9930 ±0.0011	0.9942 ±0.0008	0.9953 ±0.0007
scGen	0.9552 ±0.0036	0.9648 ±0.0027	0.9720 ±0.0020	0.9903 ±0.0011	0.9919 ±0.0009	0.9934 ±0.0008
CPA	0.8896 ±0.0057	0.9067 ±0.0043	0.9206 ±0.0033	0.9363 ±0.0076	0.9536 ±0.0052	0.9654 ±0.0046
stVAE	0.9763 ±0.0012	0.9802 ±0.0007	0.9829 ±0.0005	0.9861 ±0.0018	0.9890 ±0.0014	0.9913 ±0.0011
PerturbedVAE	0.9403 ±0.0031	0.9485 ±0.0022	0.9547 ±0.0017	0.9928 ±0.0013	0.9939 ±0.0010	0.9947 ±0.0010
PerturbEnergy-Lang	**0.9816** ±0.0009	**0.9843** ±0.0006	**0.9863** ±0.0005	0.9931 ±0.0010	0.9942 ±0.0008	0.9954 ±0.0006
PerturbEnergy-Mis	0.9792 ±0.0010	0.9823 ±0.0007	0.9844 ±0.0005	0.9935 ±0.0009	0.9945 ±0.0007	0.9956 ±0.0006

Table 9: R^2 (↑) over the true top-K DEGs. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	Replogle			Tahoe100M		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
Linear	0.9142 ±0.0050	0.9297 ±0.0037	0.9383 ±0.0031	0.7456 ±0.0108	0.8009 ±0.0078	0.8497 ±0.0051
Linear (scGPT)	0.9295 ±0.0044	0.9456 ±0.0031	0.9546 ±0.0025	0.8303 ±0.0069	0.8621 ±0.0051	0.8930 ±0.0034
CellFlow	0.9444 ±0.0035	0.9569 ±0.0025	0.9646 ±0.0020	0.8185 ±0.0074	0.8556 ±0.0053	0.8896 ±0.0035
Squidiff	0.9466 ±0.0028	0.9560 ±0.0019	0.9608 ±0.0016	0.7862 ±0.0090	0.8028 ±0.0072	0.8323 ±0.0049
PerturbDiff	0.9177 ±0.0054	0.9374 ±0.0039	0.9488 ±0.0032	0.8482 ±0.0071	0.8767 ±0.0052	0.9032 ±0.0036
STATE	0.9492 ±0.0035	0.9607 ±0.0024	0.9677 ±0.0019	0.8642 ±0.0045	0.8888 ±0.0031	0.9126 ±0.0021
scGen	0.9356 ±0.0040	0.9495 ±0.0030	0.9583 ±0.0023	0.7520 ±0.0119	0.8041 ±0.0088	0.8503 ±0.0058
CPA	0.8544 ±0.0092	0.8895 ±0.0068	0.9107 ±0.0056	0.4470 ±0.0115	0.4937 ±0.0080	0.5710 ±0.0062
stVAE	0.9436 ±0.0040	0.9584 ±0.0026	0.9664 ±0.0020	0.8690 ±0.0055	0.8918 ±0.0040	0.9143 ±0.0027
PerturbedVAE	0.9449 ±0.0039	0.9572 ±0.0027	0.9644 ±0.0023	0.8025 ±0.0092	0.8378 ±0.0068	0.8718 ±0.0047
PerturbEnergy-Lang	0.9564 ±0.0033	0.9667 ±0.0022	0.9725 ±0.0017	**0.8826** ±0.0046	**0.9035** ±0.0034	**0.9237** ±0.0023
PerturbEnergy-Mis	**0.9606** ±0.0028	**0.9697** ±0.0019	**0.9747** ±0.0015	0.8819 ±0.0046	0.9033 ±0.0033	0.9237 ±0.0023

Method	PBMC			Norman2019		
	Top-50	Top-100	Top-200	Top-50	Top-100	Top-200
Linear	0.8485 ±0.0085	0.8804 ±0.0059	0.9009 ±0.0043	0.8375 ±0.0289	0.8975 ±0.0166	0.9321 ±0.0113
Linear (scGPT)	0.8757 ±0.0074	0.9067 ±0.0050	0.9265 ±0.0036	0.8852 ±0.0029	0.9288 ±0.0019	0.9711 ±0.0014
CellFlow	0.9029 ±0.0056	0.9296 ±0.0032	0.9462 ±0.0021	0.9706 ±0.0090	0.9802 ±0.0053	0.9855 ±0.0038
Squidiff	0.8061 ±0.0097	0.8399 ±0.0066	0.8631 ±0.0048	0.9647 ±0.0037	0.9714 ±0.0027	0.9760 ±0.0022
PerturbDiff	0.8972 ±0.0086	0.9183 ±0.0061	0.9344 ±0.0035	**0.9863** ±0.0019	**0.9895** ±0.0014	**0.9918** ±0.0010
STATE	0.8833 ±0.0061	0.9145 ±0.0038	0.9339 ±0.0026	0.9803 ±0.0038	0.9855 ±0.0025	0.9892 ±0.0017
scGen	0.8874 ±0.0074	0.9163 ±0.0054	0.9359 ±0.0039	0.9735 ±0.0038	0.9805 ±0.0026	0.9852 ±0.0020
CPA	0.7335 ±0.0122	0.7857 ±0.0089	0.8224 ±0.0067	0.8174 ±0.0296	0.8845 ±0.0169	0.9235 ±0.0115
stVAE	0.9302 ±0.0035	0.9470 ±0.0021	0.9576 ±0.0014	0.9592 ±0.0066	0.9716 ±0.0045	0.9800 ±0.0029
PerturbedVAE	0.8503 ±0.0072	0.8791 ±0.0047	0.8957 ±0.0035	0.9785 ±0.0056	0.9840 ±0.0035	0.9877 ±0.0026
PerturbEnergy-Lang	**0.9434** ±0.0036	**0.9570** ±0.0020	**0.9647** ±0.0015	0.9803 ±0.0033	0.9854 ±0.0023	0.9893 ±0.0015
PerturbEnergy-Mis	0.9396 ±0.0035	0.9536 ±0.0022	0.9615 ±0.0015	0.9818 ±0.0030	0.9864 ±0.0021	0.9899 ±0.0014

Table 10: HVG-group correlation-based measures on Replogle, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups; the perturbation-discrimination scores are computed once over the whole test set and carry no such spread. Best in **bold**, second best underlined.

Method	PCC ↑	R^2 ↑	PDCorr ↑	LFCSpear ↑	LogFC-cos ↑
Linear	0.9772 ±0.0011	0.9529 ±0.0022	0.3323 ±0.0129	0.2519 ±0.0117	0.3368 ±0.0131
Linear (scGPT)	0.9868 ±0.0006	0.9732 ±0.0013	0.4584 ±0.0175	0.3607 ±0.0145	0.4688 ±0.0174
CellFlow	0.9706 ±0.0012	0.9393 ±0.0024	0.5559 ±0.0187	0.4623 ±0.0170	0.5648 ±0.0185
Squidiff	0.9829 ±0.0005	0.9649 ±0.0009	0.4945 ±0.0170	0.4536 ±0.0141	0.5065 ±0.0169
PerturbDiff	0.9875 ±0.0008	0.9743 ±0.0015	0.5470 ±0.0174	0.5186 ±0.0137	0.5557 ±0.0173
STATE	0.9905 ±0.0005	0.9805 ±0.0010	0.5091 ±0.0206	0.4323 ±0.0183	0.5177 ±0.0205
scGen	0.9883 ±0.0006	0.9760 ±0.0012	0.5381 ±0.0138	0.4900 ±0.0118	0.5411 ±0.0139
CPA	0.9815 ±0.0012	0.9614 ±0.0025	0.3039 ±0.0062	0.3040 ±0.0047	0.3101 ±0.0063
stVAE	0.9903 ±0.0005	0.9802 ±0.0010	0.5648 ±0.0158	0.5142 ±0.0132	0.5715 ±0.0158
PerturbedVAE	0.9902 ±0.0005	0.9795 ±0.0011	0.5578 ±0.0151	0.5019 ±0.0128	0.5600 ±0.0152
PerturbEnergy-Lang	**0.9915** ±0.0004	**0.9825** ±0.0009	0.6000 ±0.0154	0.5470 ±0.0131	0.6070 ±0.0154
PerturbEnergy-Mis	0.9913 ±0.0004	0.9821 ±0.0008	**0.6038** ±0.0165	**0.5492** ±0.0138	**0.6106** ±0.0164

Method	DirAgr ↑	PDS-cos ↓	PDS-ℓ_1 ↓	PDS-ℓ_2 ↓
Linear	0.5780 ±0.0043	0.2465	0.2992	0.2790
Linear (scGPT)	0.6146 ±0.0046	0.3852	0.3542	0.3512
CellFlow	0.5430 ±0.0024	0.5042	0.4987	0.5012
Squidiff	0.6645 ±0.0048	0.3889	0.2657	0.2493
PerturbDiff	0.6903 ±0.0047	0.4208	0.4477	0.4437
STATE	0.6456 ±0.0064	0.3266	0.2150	0.2156
scGen	0.6673 ±0.0042	0.2336	0.3026	0.2868
CPA	0.6038 ±0.0015	0.3573	0.4960	0.4967
stVAE	0.6817 ±0.0047	0.3461	0.2521	0.2550
PerturbedVAE	0.6756 ±0.0046	**0.1744**	0.2383	0.2326
PerturbEnergy-Lang	0.6938 ±0.0047	0.2633	**0.1939**	**0.1902**
PerturbEnergy-Mis	**0.6973** ±0.0049	0.3329	0.1971	0.1932

Table 11: HVG-group DEG-related scores on Replogle, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	DEPrec ↑	DERec ↑	DEOverlap ↑	AUROC ↑	AUPRC ↑
Linear	0.2047 ±0.0075	0.2047 ±0.0075	0.1198 ±0.0049	0.7200 ±0.0076	0.1662 ±0.0073
Linear (scGPT)	0.2590 ±0.0106	0.2590 ±0.0106	0.1615 ±0.0073	0.7698 ±0.0092	0.2409 ±0.0115
CellFlow	0.0967 ±0.0033	0.0967 ±0.0033	0.0518 ±0.0019	0.6708 ±0.0054	0.0806 ±0.0023
Squidiff	0.2566 ±0.0104	0.2566 ±0.0104	0.1592 ±0.0071	0.7981 ±0.0089	0.2475 ±0.0114
PerturbDiff	0.2982 ±0.0132	0.2982 ±0.0132	0.1973 ±0.0100	0.8096 ±0.0085	0.2938 ±0.0147
STATE	0.3049 ±0.0134	0.3049 ±0.0134	0.2024 ±0.0101	0.7946 ±0.0101	0.2997 ±0.0149
scGen	0.2739 ±0.0098	0.2739 ±0.0098	0.1696 ±0.0068	0.8067 ±0.0079	0.2537 ±0.0103
CPA	0.1215 ±0.0041	0.1215 ±0.0041	0.0661 ±0.0024	0.6707 ±0.0045	0.0859 ±0.0028
stVAE	0.2837 ±0.0126	0.2837 ±0.0126	0.1848 ±0.0093	0.7987 ±0.0089	0.2713 ±0.0138
PerturbedVAE	0.2915 ±0.0114	0.2915 ±0.0114	0.1863 ±0.0082	0.8075 ±0.0084	0.2805 ±0.0128
PerturbEnergy-Lang	**0.3226** ±0.0129	**0.3226** ±0.0129	**0.2140** ±0.0098	**0.8233** ±0.0086	**0.3173** ±0.0147
PerturbEnergy-Mis	0.3219 ±0.0131	0.3219 ±0.0131	0.2139 ±0.0100	0.8202 ±0.0089	0.3164 ±0.0149

Table 12: HVG-group distribution-level distances on Replogle, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	RMSE ↓	MSE ↓	MAE ↓	EDist ↓	Sinkhorn ↓
Linear	0.1122 ±0.0025	0.0144 ±0.0007	0.0764 ±0.0016	1.4655 ±0.0592	1.0269 ±0.0047
Linear (scGPT)	0.0860 ±0.0017	0.0082 ±0.0004	0.0585 ±0.0010	0.9934 ±0.0352	0.9918 ±0.0035
CellFlow	0.1303 ±0.0024	0.0186 ±0.0007	0.0866 ±0.0015	4.4345 ±0.0812	0.6073 ±0.0052
Squidiff	0.1021 ±0.0012	0.0108 ±0.0003	0.0656 ±0.0007	3.7791 ±0.0379	0.5664 ±0.0035
PerturbDiff	0.0817 ±0.0020	0.0079 ±0.0005	0.0527 ±0.0012	6.3453 ±0.0534	0.5637 ±0.0041
STATE	0.0725 ±0.0016	0.0060 ±0.0003	0.0491 ±0.0009	**0.7007** ±0.0274	0.9903 ±0.0038
scGen	0.0795 ±0.0019	0.0073 ±0.0004	0.0525 ±0.0011	3.4475 ±0.0548	0.5432 ±0.0036
CPA	0.0953 ±0.0030	0.0118 ±0.0008	0.0612 ±0.0018	4.8973 ±0.0860	0.5575 ±0.0052
stVAE	0.0738 ±0.0015	0.0061 ±0.0003	0.0490 ±0.0008	5.8831 ±0.0569	**0.5376** ±0.0038
PerturbedVAE	0.0745 ±0.0016	0.0063 ±0.0003	0.0501 ±0.0009	5.5135 ±0.0444	0.5545 ±0.0038
PerturbEnergy-Lang	**0.0691** ±0.0014	**0.0054** ±0.0003	**0.0465** ±0.0009	8.6506 ±0.0427	0.5577 ±0.0037
PerturbEnergy-Mis	0.0704 ±0.0014	0.0055 ±0.0002	0.0473 ±0.0008	9.0347 ±0.0442	0.5587 ±0.0036

Table 13: HVG-group correlation-based measures on PBMC, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups; the perturbation-discrimination scores are computed once over the whole test set and carry no such spread. Best in **bold**, second best underlined.

Method	PCC ↑	R^2 ↑	PDCorr ↑	LFCSpear ↑	LogFC-cos ↑
Linear	0.9705 ±0.0009	0.9372 ±0.0020	0.2015 ±0.0123	0.0708 ±0.0060	0.1960 ±0.0123
Linear (scGPT)	0.9826 ±0.0007	0.9636 ±0.0014	0.2922 ±0.0119	0.0881 ±0.0049	0.2934 ±0.0119
CellFlow	0.9784 ±0.0013	0.9555 ±0.0025	0.2301 ±0.0067	0.2610 ±0.0051	0.2294 ±0.0069
Squidiff	0.9605 ±0.0009	0.8978 ±0.0016	0.3061 ±0.0102	0.2693 ±0.0044	0.3003 ±0.0103
PerturbDiff	0.9805 ±0.0007	0.9521 ±0.0021	0.4435 ±0.0125	0.3566 ±0.0057	0.4429 ±0.0126
STATE	0.9868 ±0.0006	0.9724 ±0.0012	0.3074 ±0.0140	0.0808 ±0.0061	0.3078 ±0.0140
scGen	0.9857 ±0.0007	0.9706 ±0.0013	0.4126 ±0.0118	0.3594 ±0.0047	0.4121 ±0.0118
CPA	0.9425 ±0.0016	0.8794 ±0.0033	0.1275 ±0.0075	0.2351 ±0.0029	0.1207 ±0.0078
stVAE	0.9887 ±0.0003	0.9760 ±0.0006	0.4886 ±0.0119	0.3531 ±0.0050	0.4859 ±0.0119
PerturbedVAE	0.9654 ±0.0009	0.9232 ±0.0015	0.3244 ±0.0151	0.3069 ±0.0056	0.3241 ±0.0151
PerturbEnergy-Lang	**0.9903** ±0.0003	**0.9781** ±0.0006	**0.5577** ±0.0103	**0.3689** ±0.0045	**0.5562** ±0.0103
PerturbEnergy-Mis	0.9884 ±0.0003	0.9746 ±0.0007	0.5345 ±0.0108	0.3499 ±0.0048	0.5311 ±0.0109

Method	DirAgr ↑	PDS-cos ↓	PDS-ℓ_1 ↓	PDS-ℓ_2 ↓
Linear	0.5181 ±0.0021	0.2185	0.3569	0.3141
Linear (scGPT)	0.5265 ±0.0016	0.1139	0.3067	0.2689
CellFlow	0.6100 ±0.0021	0.1190	0.2487	0.3699
Squidiff	0.6191 ±0.0016	0.1116	0.1344	0.1836
PerturbDiff	0.6552 ±0.0020	0.0920	0.0929	0.1461
STATE	0.5247 ±0.0018	0.1591	0.3306	0.2800
scGen	**0.6556** ±0.0019	0.0689	0.1641	0.2173
CPA	0.5959 ±0.0012	0.2560	0.1856	0.3423
stVAE	0.6418 ±0.0017	0.0432	0.0777	0.1181
PerturbedVAE	0.6293 ±0.0021	0.1709	0.1135	0.1980
PerturbEnergy-Lang	0.6376 ±0.0016	**0.0289**	**0.0463**	**0.0681**
PerturbEnergy-Mis	0.6325 ±0.0016	0.0317	0.0541	0.0706

Table 14: HVG-group DEG-related scores on PBMC, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	DEPrec ↑	DERec ↑	DEOverlap ↑	AUROC ↑	AUPRC ↑
Linear	0.1404 ±0.0055	0.1404 ±0.0055	0.0786 ±0.0035	0.7303 ±0.0045	0.1169 ±0.0059
Linear (scGPT)	0.1693 ±0.0067	0.1693 ±0.0067	0.0975 ±0.0046	0.7147 ±0.0052	0.1469 ±0.0070
CellFlow	0.1349 ±0.0035	0.1349 ±0.0035	0.0735 ±0.0021	0.7689 ±0.0030	0.1062 ±0.0024
Squidiff	0.1501 ±0.0063	0.1501 ±0.0063	0.0853 ±0.0041	0.8073 ±0.0036	0.1363 ±0.0058
PerturbDiff	0.1995 ±0.0076	0.1995 ±0.0076	0.1176 ±0.0054	**0.8215** ±0.0039	0.1701 ±0.0080
STATE	0.1929 ±0.0083	0.1929 ±0.0083	0.1149 ±0.0060	0.7200 ±0.0060	0.1724 ±0.0089
scGen	0.1993 ±0.0074	0.1993 ±0.0074	0.1170 ±0.0052	0.8189 ±0.0035	0.1750 ±0.0077
CPA	0.1392 ±0.0046	0.1392 ±0.0046	0.0769 ±0.0028	0.7787 ±0.0039	0.1124 ±0.0036
stVAE	0.2208 ±0.0092	0.2208 ±0.0092	0.1347 ±0.0069	0.8173 ±0.0044	0.2077 ±0.0105
PerturbedVAE	0.1709 ±0.0073	0.1709 ±0.0073	0.0994 ±0.0050	0.8155 ±0.0037	0.1562 ±0.0076
PerturbEnergy-Lang	**0.2473** ±0.0090	**0.2473** ±0.0090	**0.1515** ±0.0068	0.8205 ±0.0044	**0.2216** ±0.0105
PerturbEnergy-Mis	0.2366 ±0.0093	0.2366 ±0.0093	0.1451 ±0.0070	0.8148 ±0.0044	0.2142 ±0.0107

Table 15: HVG-group distribution-level distances on PBMC, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	RMSE ↓	MSE ↓	MAE ↓	EDist ↓	Sinkhorn ↓
Linear	0.0376 ±0.0009	0.0016 ±0.0001	0.0195 ±0.0004	0.2725 ±0.0137	0.4927 ±0.0113
Linear (scGPT)	0.0299 ±0.0010	0.0012 ±0.0001	0.0167 ±0.0004	0.2035 ±0.0137	0.4902 ±0.0111
CellFlow	0.0338 ±0.0014	0.0017 ±0.0002	0.0159 ±0.0006	1.0656 ±0.0340	0.2766 ±0.0072
Squidiff	0.0529 ±0.0013	0.0033 ±0.0002	0.0217 ±0.0006	1.6522 ±0.0339	0.2937 ±0.0077
PerturbDiff	0.0340 ±0.0008	0.0014 ±0.0001	0.0146 ±0.0004	2.4800 ±0.0287	0.2905 ±0.0071
STATE	0.0264 ±0.0009	0.0009 ±0.0001	0.0138 ±0.0005	**0.1658** ±0.0112	0.4879 ±0.0112
scGen	0.0275 ±0.0010	0.0011 ±0.0001	**0.0123** ±0.0005	0.8381 ±0.0238	0.2793 ±0.0072
CPA	0.0545 ±0.0012	0.0034 ±0.0002	0.0202 ±0.0006	4.0187 ±0.0435	0.2913 ±0.0074
stVAE	0.0251 ±0.0006	0.0007 ±0.0000	0.0125 ±0.0003	1.8705 ±0.0282	**0.2630** ±0.0063
PerturbedVAE	0.0443 ±0.0008	0.0022 ±0.0001	0.0183 ±0.0004	2.0305 ±0.0369	0.2886 ±0.0075
PerturbEnergy-Lang	**0.0239** ±0.0006	0.0007 ±0.0000	0.0127 ±0.0003	2.6597 ±0.0327	0.2698 ±0.0067
PerturbEnergy-Mis	0.0250 ±0.0005	**0.0007** ±0.0000	0.0132 ±0.0003	2.5102 ±0.0304	0.2701 ±0.0066

Table 16: HVG-group correlation-based measures on Norman2019, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups; the perturbation-discrimination scores are computed once over the whole test set and carry no such spread. Best in **bold**, second best underlined.

Method	PCC ↑	R^2 ↑	PDCorr ↑	LFCSpear ↑	LogFC-cos ↑
Linear	0.9837 ±0.0022	0.9674 ±0.0043	0.4668 ±0.0426	0.2613 ±0.0231	0.4710 ±0.0422
Linear (scGPT)	0.9971 ±0.0003	0.9941 ±0.0007	0.9051 ±0.0160	0.3811 ±0.0145	0.9055 ±0.0160
CellFlow	0.9772 ±0.0027	0.9545 ±0.0053	**0.9137** ±0.0270	0.3883 ±0.0141	**0.9141** ±0.0146
Squidiff	0.9921 ±0.0006	0.9839 ±0.0012	0.8296 ±0.0220	0.4656 ±0.0150	0.8307 ±0.0219
PerturbDiff	**0.9975** ±0.0003	**0.9944** ±0.0006	0.8985 ±0.0205	**0.5092** ±0.0132	0.8984 ±0.0205
STATE	0.9969 ±0.0004	0.9936 ±0.0008	0.9078 ±0.0153	0.3700 ±0.0164	0.9078 ±0.0153
scGen	0.9957 ±0.0004	0.9911 ±0.0010	0.8800 ±0.0155	0.4622 ±0.0117	0.8802 ±0.0154
CPA	0.9818 ±0.0023	0.9636 ±0.0045	0.3727 ±0.0216	0.2036 ±0.0122	0.3769 ±0.0216
stVAE	0.9948 ±0.0007	0.9894 ±0.0013	0.8381 ±0.0253	0.3913 ±0.0122	0.8381 ±0.0254
PerturbedVAE	0.9964 ±0.0006	0.9926 ±0.0012	0.8980 ±0.0134	0.3736 ±0.0098	0.8984 ±0.0134
PerturbEnergy-Lang	0.9970 ±0.0004	0.9936 ±0.0007	0.9063 ±0.0150	0.4280 ±0.0123	0.9066 ±0.0150
PerturbEnergy-Mis	0.9971 ±0.0003	0.9939 ±0.0007	0.9112 ±0.0141	0.4350 ±0.0122	0.9116 ±0.0141

Method	DirAgr ↑	PDS-cos ↓	PDS-ℓ_1 ↓	PDS-ℓ_2 ↓
Linear	0.6395 ±0.0120	0.5000	0.5000	0.5000
Linear (scGPT)	0.6759 ±0.0075	**0.0027**	0.0054	0.0034
CellFlow	0.6832 ±0.0072	0.0040	0.0297	0.0216
Squidiff	0.7509 ±0.0078	0.0061	0.0074	0.0054
PerturbDiff	**0.7630** ±0.0064	0.0202	**0.0047**	**0.0027**
STATE	0.6693 ±0.0090	0.0047	0.0142	0.0101
scGen	0.7342 ±0.0061	0.0088	0.0202	0.0094
CPA	0.6043 ±0.0061	0.4784	0.4946	0.4926
stVAE	0.6925 ±0.0059	0.0263	0.0897	0.0830
PerturbedVAE	0.6800 ±0.0050	0.0040	0.0155	0.0115
PerturbEnergy-Lang	0.7115 ±0.0061	0.0040	0.0108	0.0094
PerturbEnergy-Mis	0.7140 ±0.0059	0.0027	0.0088	0.0081

Table 17: HVG-group DEG-related scores on Norman2019, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	DEPrec ↑	DERec ↑	DEOverlap ↑	AUROC ↑	AUPRC ↑
Linear	0.3251 ±0.0119	0.3251 ±0.0119	0.1964 ±0.0086	0.9423 ±0.0032	0.3358 ±0.0161
Linear (scGPT)	0.6528 ±0.0203	0.6528 ±0.0203	0.4960 ±0.0201	0.9777 ±0.0053	0.7284 ±0.0251
CellFlow	0.2190 ±0.0097	0.2190 ±0.0097	0.1242 ±0.0062	0.8502 ±0.0071	0.2118 ±0.0089
Squidiff	0.5682 ±0.0213	0.5682 ±0.0213	0.4079 ±0.0197	0.9735 ±0.0035	0.6335 ±0.0252
PerturbDiff	0.6467 ±0.0217	0.6467 ±0.0217	0.4909 ±0.0216	**0.9809** ±0.0037	0.7091 ±0.0268
STATE	0.6523 ±0.0210	0.6523 ±0.0210	0.4963 ±0.0209	0.9757 ±0.0064	0.7227 ±0.0260
scGen	0.5969 ±0.0213	0.5969 ±0.0213	0.4369 ±0.0200	0.9732 ±0.0046	0.6618 ±0.0266
CPA	0.2759 ±0.0135	0.2759 ±0.0135	0.1628 ±0.0094	0.8958 ±0.0055	0.2699 ±0.0131
stVAE	0.5774 ±0.0228	0.5774 ±0.0228	0.4184 ±0.0207	0.9691 ±0.0051	0.6296 ±0.0277
PerturbedVAE	0.6513 ±0.0203	0.6513 ±0.0203	0.4942 ±0.0199	0.9747 ±0.0058	0.7118 ±0.0254
PerturbEnergy-Lang	0.6554 ±0.0202	0.6554 ±0.0202	0.4988 ±0.0201	0.9776 ±0.0053	0.7215 ±0.0245
PerturbEnergy-Mis	**0.6636** ±0.0203	**0.6636** ±0.0203	**0.5083** ±0.0205	0.9804 ±0.0043	**0.7309** ±0.0234

Table 18: HVG-group distribution-level distances on Norman2019, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	RMSE ↓	MSE ↓	MAE ↓	EDist ↓	Sinkhorn ↓
Linear	0.0762 ±0.0048	0.0067 ±0.0009	0.0203 ±0.0014	1.2427 ±0.1400	1.3307 ±0.0529
Linear (scGPT)	0.0325 ±0.0019	0.0012 ±0.0001	0.0132 ±0.0006	0.5972 ±0.0622	1.1605 ±0.0364
CellFlow	0.0352 ±0.0029	0.0016 ±0.0003	0.0113 ±0.0010	**0.2976** ±0.0452	1.2613 ±0.0464
Squidiff	0.0551 ±0.0022	0.0032 ±0.0002	0.0174 ±0.0007	0.9530 ±0.0515	1.0328 ±0.0339
PerturbDiff	**0.0318** ±0.0017	**0.0011** ±0.0001	**0.0105** ±0.0006	2.8086 ±0.1059	**0.8198** ±0.0293
STATE	0.0337 ±0.0021	0.0013 ±0.0002	0.0127 ±0.0007	0.3919 ±0.0490	1.2014 ±0.0364
scGen	0.0400 ±0.0022	0.0018 ±0.0002	0.0129 ±0.0008	0.9353 ±0.0814	0.9480 ±0.0348
CPA	0.0803 ±0.0051	0.0074 ±0.0010	0.0229 ±0.0015	5.0856 ±0.2118	0.9943 ±0.0549
stVAE	0.0434 ±0.0026	0.0021 ±0.0003	0.0142 ±0.0008	2.8448 ±0.1561	0.8282 ±0.0415
PerturbedVAE	0.0356 ±0.0025	0.0015 ±0.0003	0.0148 ±0.0009	3.5358 ±0.1081	0.8481 ±0.0394
PerturbEnergy-Lang	0.0339 ±0.0019	0.0013 ±0.0002	0.0126 ±0.0006	3.6007 ±0.1238	0.8649 ±0.0376
PerturbEnergy-Mis	0.0330 ±0.0019	0.0012 ±0.0001	0.0123 ±0.0006	4.1930 ±0.1199	0.8724 ±0.0384

Table 19: HVG-group correlation-based measures on Tahoe100M, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups; the perturbation-discrimination scores are computed once over the whole test set and carry no such spread. Best in **bold**, second best underlined.

Method	PCC ↑	R^2 ↑	PDCorr ↑	LFCSpear ↑	LogFC-cos ↑
Linear	0.9761 ±0.0006	0.9451 ±0.0018	0.3223 ±0.0057	0.1999 ±0.0040	0.3344 ±0.0065
Linear (scGPT)	0.9783 ±0.0005	0.9532 ±0.0012	0.4708 ±0.0076	0.2459 ±0.0038	0.4803 ±0.0076
CellFlow	0.9804 ±0.0005	0.9583 ±0.0012	0.5560 ±0.0055	0.2803 ±0.0033	0.5630 ±0.0056
Squidiff	0.9343 ±0.0009	0.8561 ±0.0023	0.3047 ±0.0050	0.1878 ±0.0027	0.3067 ±0.0049
PerturbDiff	0.9798 ±0.0007	0.9529 ±0.0013	0.5307 ±0.0067	0.3664 ±0.0033	0.5192 ±0.0074
STATE	0.9815 ±0.0004	**0.9613** ±0.0008	0.5694 ±0.0063	0.2884 ±0.0037	0.5755 ±0.0065
scGen	0.9726 ±0.0007	0.9380 ±0.0021	0.2986 ±0.0042	0.2375 ±0.0030	0.3022 ±0.0043
CPA	0.8190 ±0.0017	0.6587 ±0.0028	0.2654 ±0.0096	0.2175 ±0.0042	0.2588 ±0.0103
stVAE	0.9794 ±0.0005	0.9564 ±0.0010	0.5456 ±0.0056	0.3420 ±0.0028	0.5506 ±0.0059
PerturbedVAE	0.9707 ±0.0006	0.9341 ±0.0017	0.3686 ±0.0042	0.2777 ±0.0023	0.3720 ±0.0044
PerturbEnergy-Lang	0.9818 ±0.0004	0.9605 ±0.0010	0.5944 ±0.0056	0.3680 ±0.0030	0.6003 ±0.0058
PerturbEnergy-Mis	**0.9820** ±0.0004	0.9610 ±0.0010	**0.5974** ±0.0055	**0.3774** ±0.0029	**0.6031** ±0.0057

Method	DirAgr ↑	PDS-cos ↓	PDS-ℓ_1 ↓	PDS-ℓ_2 ↓
Linear	0.5783 ±0.0020	0.0730	0.2357	0.2619
Linear (scGPT)	0.5879 ±0.0017	0.0851	0.1053	0.1340
CellFlow	0.5952 ±0.0015	0.0404	0.1137	0.1306
Squidiff	0.5719 ±0.0014	0.1401	0.1434	0.1548
PerturbDiff	**0.6568** ±0.0015	0.1439	0.1699	0.1771
STATE	0.6036 ±0.0016	**0.0380**	**0.0378**	**0.0389**
scGen	0.6021 ±0.0013	0.1002	0.2124	0.2616
CPA	0.6106 ±0.0016	0.2641	0.2268	0.2642
stVAE	0.6354 ±0.0011	0.0591	0.0831	0.0816
PerturbedVAE	0.6141 ±0.0010	0.0887	0.1507	0.1798
PerturbEnergy-Lang	0.6457 ±0.0013	0.0454	0.0536	0.0587
PerturbEnergy-Mis	0.6485 ±0.0013	0.0469	0.0540	0.0608

Table 20: HVG-group DEG-related scores on Tahoe100M, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	DEPrec ↑	DERec ↑	DEOverlap ↑	AUROC ↑	AUPRC ↑
Linear	0.1318 ±0.0038	0.1318 ±0.0038	0.0720 ±0.0023	0.7426 ±0.0034	0.1015 ±0.0030
Linear (scGPT)	0.2070 ±0.0059	0.2070 ±0.0059	0.1192 ±0.0039	0.7713 ±0.0040	0.1730 ±0.0057
CellFlow	0.2339 ±0.0056	0.2339 ±0.0056	0.1361 ±0.0038	0.7950 ±0.0036	0.1972 ±0.0059
Squidiff	0.1341 ±0.0035	0.1341 ±0.0035	0.0731 ±0.0021	0.7502 ±0.0029	0.0954 ±0.0026
PerturbDiff	0.2054 ±0.0054	0.2054 ±0.0054	0.1176 ±0.0035	0.7751 ±0.0039	0.1659 ±0.0051
STATE	0.2407 ±0.0061	0.2407 ±0.0061	0.1411 ±0.0042	0.7951 ±0.0039	0.2042 ±0.0065
scGen	0.1311 ±0.0028	0.1311 ±0.0028	0.0709 ±0.0016	0.7374 ±0.0032	0.0922 ±0.0019
CPA	0.1353 ±0.0044	0.1353 ±0.0044	0.0744 ±0.0026	0.7500 ±0.0039	0.1094 ±0.0030
stVAE	0.2209 ±0.0053	0.2209 ±0.0053	0.1274 ±0.0036	0.7790 ±0.0040	0.1786 ±0.0057
PerturbedVAE	0.1513 ±0.0039	0.1513 ±0.0039	0.0833 ±0.0024	0.7571 ±0.0036	0.1087 ±0.0029
PerturbEnergy-Lang	0.2408 ±0.0057	0.2408 ±0.0057	0.1406 ±0.0039	0.7952 ±0.0040	0.2050 ±0.0060
PerturbEnergy-Mis	**0.2452** ±0.0058	**0.2452** ±0.0058	**0.1437** ±0.0040	**0.8004** ±0.0039	**0.2072** ±0.0061

Table 21: HVG-group distribution-level distances on Tahoe100M, over all 2,000 highly variable genes. Values are the mean over held-out perturbation groups ± the s.e.m. across those groups. Best in **bold**, second best underlined.

Method	RMSE ↓	MSE ↓	MAE ↓	EDist ↓	Sinkhorn ↓
Linear	0.1243 ±0.0012	0.0159 ±0.0004	0.0632 ±0.0005	0.9646 ±0.0147	0.7265 ±0.0008
Linear (scGPT)	0.1166 ±0.0010	0.0139 ±0.0003	0.0629 ±0.0004	0.9021 ±0.0115	0.7206 ±0.0008
CellFlow	0.1093 ±0.0010	0.0122 ±0.0003	**0.0560** ±0.0004	0.8195 ±0.0103	0.7319 ±0.0013
Squidiff	0.2058 ±0.0010	0.0427 ±0.0004	0.1101 ±0.0005	6.7723 ±0.0579	0.4257 ±0.0011
PerturbDiff	0.1166 ±0.0012	0.0140 ±0.0003	0.0581 ±0.0005	21.6085 ±0.0591	0.3951 ±0.0008
STATE	0.1062 ±0.0007	**0.0114** ±0.0002	0.0560 ±0.0004	**0.7920** ±0.0073	0.7239 ±0.0008
scGen	0.1324 ±0.0013	0.0181 ±0.0004	0.0677 ±0.0005	3.7313 ±0.0280	0.4343 ±0.0010
CPA	0.3211 ±0.0021	0.1044 ±0.0013	0.1305 ±0.0007	20.7955 ±0.0703	0.4451 ±0.0010
stVAE	0.1126 ±0.0009	0.0129 ±0.0002	0.0613 ±0.0004	14.2524 ±0.0406	0.3899 ±0.0007
PerturbedVAE	0.1372 ±0.0011	0.0192 ±0.0003	0.0704 ±0.0004	7.2237 ±0.0251	**0.3887** ±0.0007
PerturbEnergy-Lang	0.1067 ±0.0009	0.0116 ±0.0002	0.0587 ±0.0004	21.1279 ±0.0236	0.3943 ±0.0007
PerturbEnergy-Mis	**0.1060** ±0.0009	0.0115 ±0.0002	0.0579 ±0.0004	21.2050 ±0.0241	0.3942 ±0.0007